

Name

cifs.upcall – Userspace upcall helper for Common Internet File System (CIFS)

SYNOPSIS

```
cifs.upcall [--trust-dns|-t] [--version|-v] [--legacy-uid|-l]
               [--krb5conf=/path/to/krb5.conf|-k /path/to/krb5.conf] [--keytab=/path/to/keytab|-K
               /path/to/keytab] [--expire|-e nsecs] {keyid}
```

DESCRIPTION

This tool is part of the cifs-utils suite.

cifs.upcall is a userspace helper program for the linux CIFS client filesystem. There are a number of activities that the kernel cannot easily do itself. This program is a callout program that does these things for the kernel and then returns the result.

cifs.upcall is generally intended to be run when the kernel calls request-key(8) for a particular key type. While it can be run directly from the command-line, it's not generally intended to be run that way.

OPTIONS

-c This option is deprecated and is currently ignored.

--no-env-probe|-E

Normally, **cifs.upcall** will probe the environment variable space of the process that initiated the upcall in order to fetch the value of **\$KRB5CCNAME**. This can assist the program with finding credential caches in non-default locations. If this option is set, then the program won't do this and will rely on finding credcaches in the default locations specified in *krb5.conf*. Note that this is never performed when the uid is 0. The default credcache location is always used when the uid is 0, regardless of the environment variable setting in the process.

--krb5conf|-k=/path/to/krb5.conf

This option allows administrators to set an alternate location for the *krb5.conf* file that **cifs.upcall** will use.

--keytab=-K=/path/to/keytab

This option allows administrators to specify a keytab file to be used. When a user has no credential cache already established, **cifs.upcall** will attempt to use this keytab to acquire them. The default is the system-wide keytab */etc/krb5.keytab*.

--trust-dns|-t

With krb5 upcalls, the name used as the host portion of the service principal defaults to the host-name portion of the UNC. This option allows the upcall program to reverse resolve the network address of the server in order to get the hostname.

This is less secure than not trusting DNS. When using this option, it's possible that an attacker could get control of DNS and trick the client into mounting a different server altogether. It's preferable to instead add server principals to the KDC for every possible hostname, but this option exists for cases where that isn't possible. The default is to not trust reverse hostname lookups in this fashion.

--legacy-uid|-l

Traditionally, the kernel has sent only a single uid= parameter to the upcall for the SPNEGO upcall that's used to determine what user's credential cache to use. This parameter is affected by the uid= mount option, which also governs the ownership of files on the mount.

Newer kernels send a creduid= option as well, which contains what uid it thinks actually owns the credentials that it's looking for. At mount time, this is generally set to the real uid of the user doing the mount. For multisession mounts, it's set to the fsuid of the mount user. Set this option if you want cifs.upcall to use the older uid= parameter instead of the creduid= parameter.

--expire|-e

Override default timeout value (600 seconds) for **dns_resolver** key.

--version|-v

Print version number and exit.

ENVIRONMENT VARIABLES

GSS_USE_PROXY="yes"

Enable usage of gssproxy for credential retrieval. This includes keytab based client initiation as well as (Resource Based) Constrained Delegation. See gssproxy-mech(8).

CONFIGURATION FOR KEYCTL

cifs.upcall is designed to be called from the kernel via the request-key callout program. This requires that request-key be told where and how to call this program. The current **cifs.upcall** program handles two different key types:

cifs.spnego

This keytype is for retrieving kerberos session keys

dns_resolver

This key type is for resolving hostnames into IP addresses. Support for this key type may eventually be deprecated (see below).

To make this program useful for CIFS, you'll need to set up entries for them in request-key.conf(5). Here's an example of an entry for each key type:

#OPERATION	TYPE	D	C	PROGRAM	ARG1	ARG2...
=====	=====	=	=	=====	=====	=====
create	cifs.spnego	*	*	/usr/sbin/cifs.upcall	%k	
create	dns_resolver	*	*	/usr/sbin/cifs.upcall	%k	

See request-key.conf(5) for more info on each field.

The keyutils package has also started including a dns_resolver handling program as well that is preferred over the one in **cifs.upcall**. If you are using a keyutils version equal to or greater than 1.5, you should use **key.dns_resolver** to handle the **dns_resolver** keytype instead of **cifs.upcall**. See key.dns_resolver(8) for more info.

SEE ALSO

request-key.conf(5), mount.cifs(8), key.dns_resolver(8)

AUTHOR

Igor Mammedov wrote the cifs.upcall program.

Jeff Layton authored this manpage.

The maintainer of the Linux CIFS VFS is Steve French.

The Linux CIFS Mailing list is the preferred place to ask questions regarding these programs.