NAME

rpc.gssd − RPCSEC_GSS daemon

SYNOPSIS

rpc.gssd [**−DfMnlvr**] [**−k** *keytab*] [**−p** *pipefsdir*] [**−d** *ccachedir*] [**−t** *timeout*] [**−R** *realm*]

INTRODUCTION

The RPCSEC_GSS protocol, defined in RFC 5403, is used to provide strong security for RPC-based protocols such as NFS.

Before exchanging RPC requests using RPCSEC_GSS, an RPC client must establish a GSS *security context*. A security context is shared state on each end of a network transport that enables GSS-API security services.

Security contexts are established using *security credentials*. A credential grants temporary access to a secure network service, much as a railway ticket grants temporary access to use a rail service.

A user typically obtains a credential by providing a password to the **kinit(1)** command, or via a PAM library at login time. A credential acquired with a *user principal* is known as a *user credential* (see **kerberos(1)** for more on principals).

For certain operations, a credential is required which represents no user, is otherwise unprivileged, and is always available. This is referred to as a *machine credential*.

Machine credentials are typically established using a *service principal*, whose encrypted password, called its *key*, is stored in a file, called a *keytab*, to avoid requiring a user prompt. A machine credential effectively does not expire because the system can renew it as needed without user intervention.

Once obtained, credentials are typically stored in local temporary files with well-known pathnames.

DESCRIPTION

To establish GSS security contexts using these credential files, the Linux kernel RPC client depends on a userspace daemon called **rpc.gssd**. The **rpc.gssd** daemon uses the rpc_pipefs filesystem to communicate with the kernel.

User Credentials

When a user authenticates using a command such as **kinit(1),** the resulting credential is stored in a file with a well-known name constructed using the user's UID.

To interact with an NFS server on behalf of a particular Kerberos-authenticated user, the Linux kernel RPC client requests that **rpc.gssd** initialize a security context with the credential in that user's credential file.

Typically, credential files are placed in */tmp*. However, **rpc.gssd** can search for credential files in more than one directory. See the description of the **-d** option for details.

Machine Credentials

A user credential is established by a user and is then shared with the kernel and **rpc.gssd**. A machine credential is established by **rpc.gssd** for the kernel when there is no user. Therefore **rpc.gssd** must already have the materials on hand to establish this credential without requiring user intervention.

rpc.gssd searches the local system's keytab for a principal and key to use to establish the machine credential. By default, **rpc.gssd** assumes the file */etc/krb5.keytab* contains principals and keys that can be used to obtain machine credentials.

rpc.gssd searches in the following order for a principal to use. The first matching credential is used. For the search, <hostname> and <REALM> are replaced with the local system's hostname and Kerberos realm.

<HOSTNAME>$@<REALM>
root/<hostname>@<REALM>
nfs/<hostname>@<REALM>
host/<hostname>@<REALM>
root/<anyname>@<REALM>
nfs/<anyname>@<REALM>
host/<anyname>@<REALM>

The <anyname> entries match on the service name and realm, but ignore the hostname. These can be used if a principal matching the local host's name is not found.

Note that the first principal in the search order is a user principal that enables Kerberized NFS when the local system is joined to an Active Directory domain using Samba. A password for this principal must be provided in the local system's keytab.

You can specify another keytab by using the **-k** option if */etc/krb5.keytab* does not exist or does not provide one of these principals.

Credentials for UID 0

UID 0 is a special case. By default **rpc.gssd** uses the system's machine credentials for UID 0 accesses that require GSS authentication. This limits the privileges of the root user when accessing network resources that require authentication.

Specify the **-n** option when starting **rpc.gssd** if you'd like to force the root user to obtain a user credential rather than use the local system's machine credential.

When **-n** is specified, the kernel continues to request a GSS context established with a machine credential for NFSv4 operations, such as SETCLIENTID or RENEW, that manage state. If **rpc.gssd** cannot obtain a machine credential (say, the local system has no keytab), NFSv4 operations that require machine credentials will fail.

Encryption types

A realm administrator can choose to add keys encoded in a number of different encryption types to the local system's keytab. For instance, a host/ principal might have keys for the **aes256-cts-hmac-sha1-96**, **aes128-cts-hmac-sha1-96**, **des3-cbc-sha1**, and **arcfour-hmac** encryption types. This permits **rpc.gssd** to choose an appropriate encryption type that the target NFS server supports.

These encryption types are stronger than legacy single-DES encryption types. To interoperate in environments where servers support only weak encryption types, you can restrict your client to use only single-DES encryption types by specifying the **-l** option when starting **rpc.gssd**.

OPTIONS

–D The server name passed to GSSAPI for authentication is normally the name exactly as requested. e.g. for NFS it is the server name in the "servername:/path" mount request. Only if this servername appears to be an IP address (IPv4 or IPv6) or an unqualified name (no dots) will a reverse DNS lookup will be performed to get the canoncial server name.

If **–D** is present, a reverse DNS lookup will *always* be used, even if the server name looks like a canonical name. So it is needed if partially qualified, or non canonical names are regularly used.

Using **–D** can introduce a security vulnerability, so it is recommended that **–D** not be used, and that canonical names always be used when requesting services.

-f Runs **rpc.gssd** in the foreground and sends output to stderr (as opposed to syslogd)

-n When specified, UID 0 is forced to obtain user credentials which are used instead of the local system's machine credentials.

-k *keytab*

Tells **rpc.gssd** to use the keys found in *keytab* to obtain machine credentials. The default value is */etc/krb5.keytab*.

-l When specified, restricts **rpc.gssd** to sessions to weak encryption types such as **des-cbc-crc**. This option is available only when the local system's Kerberos library supports settable encryption types.

-p *path* Tells **rpc.gssd** where to look for the rpc_pipefs filesystem. The default value is */var/lib/nfs/rpc_pipefs*.

-d *search-path*

This option specifies a colon separated list of directories that **rpc.gssd** searches for credential files. The default value is */tmp:/run/user/%U*. The literal sequence "%U" can be specified to substitue

the UID of the user for whom credentials are being searched.

-M By default, machine credentials are stored in files in the first directory in the credential directory search path (see the **-d** option). When **-M** is set, **rpc.gssd** stores machine credentials in memory instead.

-v Increases the verbosity of the output (can be specified multiple times).

-r If the RPCSEC_GSS library supports setting debug level, increases the verbosity of the output (can be specified multiple times).

-R *realm*

Kerberos tickets from this *realm* will be preferred when scanning available credentials cache files to be used to create a context. By default, the default realm, as configured in the Kerberos configuration file, is preferred.

-t *timeout*

Timeout, in seconds, for kernel GSS contexts. This option allows you to force new kernel contexts to be negotiated after *timeout* seconds, which allows changing Kerberos tickets and identities frequently. The default is no explicit timeout, which means the kernel context will live the lifetime of the Kerberos service ticket used in its creation.

-T timeout

Timeout, in seconds, to create an RPC connection with a server while establishing an authenticated gss context for a user. The default timeout is set to 5 seconds. If you get messages like "WARNING: can't create tcp rpc_clnt to server %servername% for user with uid %uid%: RPC: Remote system error - Connection timed out", you should consider an increase of this timeout.

SEE ALSO

rpc.svcgssd(8), **kerberos(1), kinit(1), krb5.conf(5)**

AUTHORS

Dug Song <dugsong@umich.edu>
Andy Adamson <andros@umich.edu>
Marius Aamodt Eriksen <marius@umich.edu>
J. Bruce Fields <bfields@umich.edu>