Open Software Foundation **S. Mullan (HP)**
Request For Comments: 92.0
January 1996

DCE INTEROPERABILITY WITH KERBEROS — FUNCTIONAL SPECIFICATION

1. INTRODUCTION

The authentication portion of the DCE Security Service is primarily based on Version 5 of the Kerberos [RFC 1510] network authentication system. For the most part, this has allowed the DCE Security Server to be able to operate as a Kerberos Key Distribution Center (KDC) for Kerberos Version 5 clients [John 95]. Today, with some restrictions, you are able to successfully use Kerberos Version 5 clients with a DCE Security Server as the KDC. However, in releases 1.0 and 1.1, DCE has not officially supported or tested interoperability with Kerberos Version 5. That support and testing is now being added to DCE 1.2.

This document discusses the requirements and functionality necessary for supporting interoperability of DCE and Kerberos Version 5 clients, including the Kerberos V5 remote utilities.

2. TARGET

This technology is being provided for customers who have Kerberos V5 clients and wish to use the DCE Security Server as a Kerberos KDC. It will also allow users with valid Kerberos V5 credentials to remotely login without exposing unencrypted passwords on the network.

3. GOALS

The primary goals of the DCE 1.2.2 and Kerberos V5 interoperability project are [see RFC 63.1]:

(a) *KDC interoperability* — Test that the DCE 1.2.2 Security Server can act as a Kerberos KDC for Kerberos V5 clients.

(b) *BSD remote utility interoperability* —

  (i) Provide DCE 1.2.2 versions of the BSD 4.4-Lite remote utilities (`rlogin`, `rlogind`, `rsh`, `rshd`) which provide authentication, encryption.

  (ii) Test that they are interoperable with the beta 4 or 5 versions of MIT Kerberos V5 BSD remote utilities.

(c) *Kerberos credential cache and keytab file compatibility* — Ensure that DCE 1.2.2 clients and Kerberos V5 clients, using a DCE 1.2.2 security server as the KDC, can share credential cache and keytab files without loss of data.

The requirements are stated more specifically in Section 6.

4. NON-GOALS

The following are *not* goals:

(a) Support interoperability of DCE and Kerberos V4 clients.

(b) Support the Kerberos V5 Application Programming Interface.

(c) Support the Kerberos V5 Administration tools.

(d) Support or test interoperability with Kerberos V5 clients using GSS-API.

(e) Provide a BSD remote copy utility (`rcp`) which interoperates with the Kerberos V5 remote utilities.

(f) Support the Kerberos V5 beta 5 password change protocol with a DCE Security Server.

(g) Be able to import a Kerberos V5 database into the DCE Security Registry.

(h) Provide Kerberos V5 compatible versions of `ftp`, or `telnet`.

(i) Merge forward the DCE sources to the most current version of MIT Kerberos V5.

Also, refer to section 10 for details of which specific versions of DCE and Kerberos V5 will be tested and supported.

5. TERMINOLOGY

(a) **Authorization**

The process of determining whether a client may use a service, which objects the client is allowed to access, and the type of access allowed for each.

(b) **Authentication**

Verifying the claimed identity of a principal.

(c) **BSD**

Berkeley Software Distribution of UNIX.

(d) **Credentials**

A ticket plus the secret session key necessary to successfully use that ticket in an authentication exchange.

(e) **DCE Security Service**

A DCE service providing authentication, authorization, and other security services. It is based on Kerberos V5, but has significant extensions and added functionality.

(f) **IETF**

Internet Engineering Task Force.

(g) **KDC**

Key Distribution Center, a Kerberos V5 network service that supplies tickets and temporary session keys.

(h) **Kerberos**

Refers to the specification of Kerberos Version 5 contained in [RFC 1510].

(i) **Kerberos Realm**

Identified by a unique name, a realm consists of a KDC, and the clients and servers registered to that KDC.

(j) **Kerberos Ticket**

A record that helps a client authenticate itself to a server; contains client's identity, session key, timestamp, and other information, all sealed using the server's secret key.

(k) **Kerberos V5 Client**

A Kerberos client which has been built against a non-DCE Kerberos V5 library.

(l) **MIT Kerberos V5**

Refers to the MIT distribution of Kerberos V5. Thus, this refers to the components as implemented in that offering.

(m) **TGT**

A Kerberos V5 Ticket-Granting Ticket which can be used to obtain additional tickets to servers.

6. REQUIREMENTS

The specific requirements for this project are as follows:

(a) Provide and test Kerberos V5 and DCE client coexistence on a single machine with a DCE KDC, sharing the same credential and keytab files.

(b) Provide `rlogin` and `rsh` utilities which optionally provide user authentication including forwarding of credentials to the remote host, and encryption of session data (at least if built with U.S. Domestic DCE sources). These utilities must be able to interoperate with other Kerberos V5 remote utilities. More specifically:

   (i) An `rlogin`, `rlogind`, `rsh`, and `rshd` based on the BSD Version 4.4-Lite sources. (The 4.4-Lite release of BSD was chosen because it is not encumbered, i.e., it has no licensing restrictions.)

   (ii) An `rlogin` and `rsh` which interoperate with a non-DCE Kerberos V5 `rlogind` and `rshd`, respectively.

   (iii) An `rlogin` and `rsh` which interoperate with a non-DCE Kerberos V5 KDC.

   (iv) An `rlogind` and `rshd` which interoperate with a non-DCE Kerberos V5 `rlogin` and `rsh`, respectively.

   (v) An `rlogind` and `rshd` which interoperate with a non-DCE Kerberos V5 KDC (`rlogind` and `rshd` do not directly communicate with the KDC; however, they do share a key with the KDC).

   (vi) An `rlogind` and `rshd` which support the Kerberos V5 `.k5login` access control mechanism.

   (vii) DCE configuration support for seamless operation of the remote utilities.

   (viii) An `rlogin` and `rsh` which can delegate the user's Kerberos credentials and promote them to DCE credentials (if applicable) on the remote host, thus avoiding the need to prompt for and send an unencrypted password over the network.

(c) Provide a DCE KDC that is validated to properly support Kerberos V5 beta 4 or 5 clients according to [RFC 1510], with some exceptions as described in section 7.3.

The following requirements are *not* committed, but will be provided as time and resources permit:

(a) Support and test inter-realm authentication between a DCE KDC and a non-DCE KDC with some exceptions as described in section 7.3.1.

(b) An `rlogin`, `rsh` which can authenticate to a host in a foreign realm.

(c) An `rlogind`, `rshd` which can accept and authenticate requests from a host in a foreign realm.

7. FUNCTIONAL DEFINITION

7.1. Credential Cache/Keytab File Compatibility

A DCE 1.2.2 client will be able to coexist with a Kerberos V5 client on the same machine, using a DCE Security Server as the KDC. This requires that they be able to share keytab and credential cache files located on the local machine. In order for a DCE application to share a credential cache or keytab file with a Kerberos V5 application, they must be able to write to the file in a format that is understandable by each other.

There are currently 2 versions of formats for the keytab and credential cache files.[1] Prior to DCE 1.1, the files were created in the version 1 format. Kerberos introduced the version 2 format with the 2nd beta release, but did not maintain backwards compatibility with files created using the version 1 format. This bug was eventually fixed in the 4th beta release of Kerberos.

---

1. The beta 5 release of Kerberos introduces a 3rd version; files created with this version are not compatible with DCE 1.2.2.

DCE 1.1 is partially based on Kerberos V5 beta 4, and thus can read and write files created in either format. The table below describes the compatibility of file versions. In the lefthand column is the DCE or Kerberos release (krb5b1 denoting the first beta release of Kerberos V5), and in the second and third columns are the 2 different file formats. A ‘‘Yes’’ implies that applications built with that release can read and write files in the version noted, and a ‘‘No’’ implies that it cannot.

Kerberos Credential Cache/Keytab File Compatibility

Release	krb5v1	krb5v2
DCE 1.0/krb5b1	Yes	No
Krb5b2	No	Yes
Krb5b3	No	Yes
DCE 1.{1,2}/krb5b4	Yes	Yes
Krb5b5	Yes	Yes

For DCE 1.2.2, users will be able to select which version their host will write the files in by specifying a value for the `krb5_kt_vno` and `krb5_ccache_vno` configuration variables. These values are stored in the DCE host specific data file (see `hostdata(1M)`). Upon creating a new credential cache or keytab file, DCE will read the value of the appropriate configuration variable to determine what version should be used. If the configuration variables do not exist, DCE will use the default version provided by the Kerberos release it is based on.

Depending on the value set, there will be some tradeoffs. For example, if the `krb5_ccache_vno` is set to 2, binary compatibility with DCE 1.0 applications may be affected, since DCE 1.0 cannot parse the version 2 format. If the `krb5_ccache_vno` is set to 1, any Kerberos V5 applications based on beta 2 or 3 will not be able to parse a credential cache file created by a DCE application. Users should set these variables to try to maximize their interoperability requirements. DCE 1.2.2 utilities (`kinit, klist, kdestroy`) will be able to refresh, read, or destroy credential files created in either version 1 or 2.

There will be a `dcecp` command to allow users to display or change the configuration variables; see section 8.6.

7.2. Kerberos V5 Configuration File Compatibility

Prior to the beta 5 release of Kerberos V5, configuration of realms and KDC servers were specified in the files `/krb5/krb.realms` and `/krb5/krb.conf`. The beta 5 release of Kerberos V5 combined these files into one file located in `/etc/krb5.conf`. The files are not compatible, and since DCE is not based on Kerberos V5 beta 5, it does not understand the `/etc/krb5.conf` format.

If the beta 5 release of Kerberos V5 is compiled with the preprocessor macro `OLD_CONFIG_FILE` defined, then the old configuration files will be used instead of `/etc/krb5.conf`. This should be considered if the Kerberos V5 beta 5 applications will be run on a DCE client using a DCE Security Server. The `/krb5/krb.conf` file is initialized during the configuration of the DCE client. The `/krb5/krb.conf` is periodically checked for stale configuration and updated by the security validation service (`secval(1m)`) of `dced(1m)`.

If the `/etc/krb5.conf` file is used for configuration instead, the user or administrator will be responsible for setting up and maintaining the file so that the Kerberos V5 applications can use the DCE Security Server as a KDC.

7.3. KDC Interoperability

Section 9 of RFC 1510 defines the interoperability requirements which must be supported by all Kerberos V5 implementations. For 1.2.2, we are only supporting KDC interoperability. Each requirement is noted below with a justification of whether or not it will be supported and tested for 1.2.2.

(a) *Encryption and checksum methods.*

RFC 1510 states that the `DES-CBC-MD5` encryption mechanism must be supported. However, the current release of Kerberos V5 (beta 5) does not conform to this requirement, so this will not be supported.

RFC 1510 states that the following checksum methods must be supported: `CRC-32`, `DES-MAC`, `DES-MAC-K`, `DES-MD5`. We will only support and test `CRC-32`, since supporting the others would require a full merge of DCE to the latest version of Kerberos V5.

(b) *Realm Names.*

RFC 1510 states that all implementations must understand hierarchical realms in both the Internet Domain and the X.500 style. RFC 1510 also states that the KDC must be able to determine the names of the intermediate realms between the KDC's realm and the requested realm when a ticket granting ticket for an unknown realm is requested. The DCE 1.2.2 Security Server will recognize realm names in the Internet Domain style, but any functionality requiring hierachical trust between a DCE cell and a Kerberos V5 realm will not be tested or supported.

See Section 7.3.1 for a more detailed description of the plans for inter-realm communication between a non-DCE KDC and a DCE KDC.

(c) *Transited field encoding.*

RFC 1510 states that `DOMAIN-X500-COMPRESS` must be supported. Hierachical/transitive trust between a DCE cell and a Kerberos V5 realm will not be supported; thus the transited field encoding algorithms will not be tested.

(d) *Pre-authentication methods.*

RFC 1510 states that the `PA-TGS-REQ` preauthentication method must be supported. This will be supported by the DCE 1.2.2 Security Server and will be tested via the Ticket-Granting Service (TGS) Exchange.

RFC 1510 allows the KDC to optionally support the `PA-ENC-TIMESTAMP` preauthentication protocol. The DCE 1.2.2 Security Server will support the `PA-ENC-TIMESTAMP` preauthentication protocol, and this will be tested.

(e) *Mutual authentication.*

Mutual authentication (via the `KRB_AP_REP` message) will be tested and supported by the DCE 1.2.2 Security Server. Although the server involved in mutual authentication does not directly communicate with the KDC, it shares a key with the KDC which is used in the mutual authentication protocol between the client and server.

(f) *Ticket addresses and flags.*

Forwardable and forwarded tickets will be supported by the DCE 1.2.2 Security Server and are required to meet the remote utility requirements.

RFC 1510 states that all KDC's must pass on tickets that carry no addresses. This is supported by the DCE 1.2.2 Security Server and will be tested.

RFC 1510 states that proxiable tickets must be supported. The DCE 1.2.2 Security Server will support proxiable tickets.

RFC 1510 states that all implementations must recognize renewable and postdated tickets but need not actually implement them. The DCE 1.2.2 Security Server will support renewable and postdated tickets and this will be tested.

(g) *User-to-user authentication.*

User-to-user authentication (via the `ENC-TKT-IN-SKEY` KDC option) will be tested and supported by the DCE 1.2.2 Security Server.

(h) *Authorization data.*

RFC 1510 states that all implementations must pass all authorization data subfields from ticket-granting tickets to any derivative tickets unless directed to suppress a subfield as part of the definition of that registered subfield type.

This will be tested and supported by the DCE 1.2.2 Security Server.

Tests will be written to validate the requirements above. These tests will be built against the beta 5 release of MIT Kerberos V5 and tested with a DCE 1.2.2 Security Server acting as the Kerberos KDC. Also, manual testing will be performed with at least 2 Kerberos V5 products based on the beta 4 release of Kerberos V5.

7.3.1. Inter-realm KDC interoperability

Inter-realm support between a DCE cell and a Kerberos V5 realm is not a committed requirement for DCE 1.2.2. However, we will attempt to implement the features described below as time and resources permit.

Communication between a DCE cell and a Kerberos V5 realm is important in order to fully utilize the interoperability features of DCE 1.2.2 and Kerberos V5 clients. For DCE 1.2.2, the following inter-realm functionality may be supported:

(a) A client principal in a DCE cell will be able to request a service from a server principal in a Kerberos V5 realm. Conversely, a client principal in a Kerberos V5 realm will be able to request a service from a server principal in a DCE cell. This requires establishing trusted principal pairs in the DCE cell and the Kerberos V5 realm. See section 7.3.1.1 for more specific details.

(b) Only peer-to-peer inter-realm trust between a DCE cell and a Kerberos V5 realm may be supported. Hierachical trust other than peer-to-peer will not be supported.

(c) DCE `kinit`, `klist` and `kdestroy` can interoperate with a non-DCE Kerberos V5 KDC. For example, users will be able to authenticate as a principal in a remote Kerberos V5 realm from a DCE client using the DCE `kinit` command. In addition, the DCE `klist` and `kdestroy` commands can be used to list or destroy the user's credentials obtained from the remote realm.

7.3.1.1. Inter-realm authentication by trust peers

By registering the Ticket-Granting Service of each realm as a principal in the other realm, inter-realm authentication can be requested by principals in two different Kerberos V5 realms. For example, the following TGS principals and keys need to be created in order for principals in realm 1 and realm 2 to authenticate to each other:

```
realm1                          realm2
------                          ------
krbtgt/realm2@realm1{keyA}      krbtgt/realm1@realm2{keyB}
krbtgt/realm1@realm2{keyB}      krbtgt/realm2@realm1{keyA}
```

If a client in realm 1 needs to authenticate to a server in realm 2, the TGS request is accompanied by a TGT encrypted in the key of the remote TGS principal `krbtgt/realm2@realm1`. The foreign KDC recognizes the request to be inter-realm and decrypts it with the trusted inter-realm key and returns a ticket to the requested foreign server. If inter-realm authentication is only required in one direction, just one of the trusted principal pairs needs to be created.

If realm 1 is a DCE cell instead of a Kerberos V5 realm, only one of the principals (`krbtgt/realm2@realm1`) can be created. The DCE registry database cannot store a principal with a foreign realm (`krbtgt/realm1@realm2`). Thus, inter-realm communication only works in one direction (DCE to Kerberos).

In order to avoid the limitations of the DCE registry, a special naming convention will be used to represent the foreign principal `krbtgt/realm1@realm2` and allow it to be stored in the DCE registry database as:

```
/.../realm1/fkrbtgt/realm2
```

or equivalently as a Kerberos V5 principal:

```
fkrbtgt/realm2@realm1
```

There are a couple of cases in which DCE will need to recognize and convert a principal from a foreign realm specified as `krbtgt/realm1@realm2` to `fkrbtgt/realm2@realm1`. When a DCE KDC determines that a TGS request is from a principal in a foreign Kerberos V5 realm, the DCE KDC will convert the server principal specified in the ticket to the format above by transposing the server and realm names and replacing the `krbtgt` client name with `fkrbtgt`. The DCE KDC will then retrieve the key stored in the DCE registry associated with this name in order to decrypt the ticket in the TGS request.

In addition, the salt used in the generation of the foreign TGS principal's DES key from the plaintext password must be identical in the DCE cell and the Kerberos V5 realm. Since the default salt is determined by the principal and realm names, it is important that the structure be the same. Before the DCE KDC generates a salt for the foreign TGS account, it will convert the name (as represented by the `krb5_principal` data structure) from `fkrbtgt/realm2@realm1` to `krbtgt/realm1@realm2` so that it will be consistent with the Kerberos V5 realm.

Since the Kerberos 5 convention is to capitalize realm names, DCE 1.2.2 will allow the realm names to be specified in capital letters:

```
/.../realm1/fkrbtgt/REALM2
/.../realm1/krbtgt/REALM2
```

However, creation of DCE cell names with capital letters will not be supported.

The following `dcecp` commands establish the necessary principals and keys in the DCE cell for inter-realm authentication to a foreign Kerberos V5 realm (appropriately named `FOREIGN-REALM`):

```
principal create krbtgt/FOREIGN-REALM
principal create fkrbtgt/FOREIGN-REALM
group add none -member krbtgt/FOREIGN-REALM
group add none -member fkrbtgt/FOREIGN-REALM
organization add none -member krbtgt/FOREIGN-REALM
organization add none -member fkrbtgt/FOREIGN-REALM
account create krbtgt/FOREIGN-REALM \
       -group none -organization none \
       -mypwd {cell_admin password} -pass {password1}
account create fkrbtgt/FOREIGN-REALM \
       -group none -organization none \
       -mypwd {cell_admin password} -pass {password2}
```

In the Kerberos V5 realm, the matching trust accounts must be created with the same passwords as specified above, using the Kerberos V5 administration tools. This relies on the ability of the administrators to securely exchange the passwords used to setup the trust accounts.

Inter-cell authentication between DCE cells will not be affected by this functionality.

7.4. Remote Utilities

A highly requested feature missing from DCE is secure remote utilities which do not expose the user's password to the network. DCE 1.2.2 will provide secure versions of the BSD remote programs `rlogin`, `rlogind`, `rsh`, and `rshd`. These remote utilities will be based on BSD 4.4-Lite and will provide the following security features (where client and server are in the same DCE cell or Kerberos realm):

(a) Mutual authentication between the `rlogin/rsh` client and `rlogind/rshd` server.

(b) Authentication of the user without exposing the password to the network. This is accomplished by verifying the user possesses a Ticket Granting Ticket and using this TGT to establish mutual authentication between the client and the remote host.

(c) Optional delegation of the user's Ticket Granting Ticket to the remote host and establishment of DCE credentials allowing the user to fully utilize their credentials on the remote host.

(d) Optional encryption of all data generated during a remote login session. (The source code for this feature is only available in domestic source, since it provides arbitrary encryption of user data, which is export controlled.)

(e) Simple authorization control via the `.k5login` access control file or the `/krb5/aname` translation database.

(f) Interoperability with Kerberos V5 remote utilities. Interoperability testing will be performed using Kerberos V5 products based on the beta 4 or 5 release from MIT.

The remote utilities will be built against the beta 5 release of Kerberos V5. The remote utilities will be very similar to the Kerberos V5 based remote utilities, in order to be interoperable with them.

7.4.1. Configuration Enhancements

`dce_config` will be enhanced to provide seamless integration of the remote utilities into the DCE environment. The following additional configuration steps will be added [Brez 95]:

(a) Add the `eklogin`, `klogin`, `ekshell`, and `kshell` ports to the `/etc/services` configuration file (see `services(4)`).

(b) Create the `host/`*`fully-qualified-hostname`* principal and keys required for mutual authentication between the remote utility client and server processes.

(c) Add an entry to `/etc/inetd.conf` to automatically start the remote utility servers when a request for remote access is made (see `inetd(1M)`).

7.4.2. Remote login

BSD Remote login consists of the `rlogin` client and the `rlogind` server. Listed below are the steps involved in authentication of the user:

(a) Mutual authentication is established between the `rlogin` client process and the `rlogind` server process via the Kerberos V5 Client/Server Authentication Exchange protocol.

(b) If requested, `rlogin` forwards the user's ticket-granting ticket to the `rlogind` server. This assumes the user possesses a ticket-granting ticket which is forwardable. A forwardable TGT can be obtained with the `-f` option of `kinit` or via the `sec_login_forwardable` flag to `sec_login_stup_identity(3)`.[2] The `rlogind` server receives the forwarded credentials and stores them in a local credential cache file.

(c) Next, the `rlogind` server checks the `.k5login` access control file and the krb5 `aname` database to determine if the user is authorized to access the remote machine. The `.k5login` access control file is located in the home directory of the account which is being remotely logged into.

(d) The `rlogind` server then promotes the forwarded credentials to DCE credentials by executing the `k5dcelogin` utility program (see section 7.3.4). If the `k5dcelogin` command fails to create a DCE login context, the errors are ignored in order to be interoperable with a non-DCE KDC (requirement (6-b-v)).

(e) The `rlogind` server executes the system login program with an option to avoid prompting for a password.

(f) If requested, all remote login session data is encrypted with the session key established for mutual authentication.

(g) If any of the steps above fail, the `rlogin` process falls back to the standard BSD 4.4-Lite `rlogin`.

See the user interface definition for `rlogin` and `rlogind` in section 8 for a complete list of options.

7.4.3. Remote shell

BSD remote shell consists of the `rsh` client and the `rshd` server. The authentication enhancements are identical to that of `rlogin` and `rlogind` with the following exceptions:

(a) There is no encryption of session data, unless the user does not specify a command to execute, in which case `rsh` executes the `rlogin` command.

---

2. The `sec_login_forwardable` functionality is not committed at this time. Refer to section 9.1 for more information.

(b) The rshd server executes the specified command instead of the system login program.

See the user interface definition for rsh and rshd in section 8 for a complete list of options.

7.4.4. Promotion of Kerberos credentials to DCE credentials

The k5dcelogin command will be used to promote Kerberos credentials to DCE credentials, so that users will be able to fully utilize DCE security features on the remote host. Listed below are the steps to accomplish this:

(a) Verify that the user possesses a ticket-granting ticket to the Authentication Service and store it in memory.

(b) Check to see if the user possesses a DCE privilege ticket-granting ticket. If found, this means that the user already has DCE credentials, and no further work is necessary; go to step (e).

(c) Destroy the user's credential cache because a fresh cache will be created by sec_login_setup_identity(3) in the next step.

(d) Setup, validate, certify and store (to disk) a DCE login context, but do not request an initial ticket-granting ticket. Instead, use the ticket-granting ticket stored in memory.

(e) Execute the system login program or the command specified by the user.

Ideally, the functionality of the k5dcelogin should be contained within the login program. However, the login path is vendor specific, thus k5dcelogin provides a solution which does not require modification of OS login source.

7.4.5. PAM integration

[RFC 86.0] discusses the Pluggable Authentication Modules architecture and how it can be integrated with OSF-DCE. Since this is not yet a DCE deliverable, the remote utilities do not need to be integrated with the PAM APIs. However, the functional definition should not preclude the integration with those APIs.

8. USER INTERFACES

8.1. rlogin

The user interface for the rlogin command, based on the BSD 4.4-Lite version, is defined below, in manpage-like fashion. Only the additional options related to authentication are described.[3]

NAME

rlogin — remote login

SYNOPSIS

```
rlogin [-8EFKLdfx] [-e char] [-k realm] [-l username] host
```

DESCRIPTION

rlogin starts a terminal session on the remote host host.

The security relevant options are as follows:

-k

The -k option requests rlogin to obtain tickets for the remote host in realm realm instead of the remote host's realm as determined by krb_realmofhost(3).

-x

The -x option turns on DES encryption for all data passed via the rogin session. This may impact response time and CPU utilization, but provides increased security. (This option is subject to

3. The -k option may not be supported: see section 7.3.1.

export control.)

`-K`

The `-K` option turns off all Kerberos authentication.

`-f`

The `-f` option requests `rlogin` to forward the user's Kerberos credentials to the remote host.

`-F`

The `-F` option causes `rlogin` to forward the user's Kerberos credentials to the remote host and to allow the remote host to forward them to subsequent remote hosts.

KERBEROS AUTHORIZATION

Each user may have a private authorization list in a file `.k5login` in their login directory. Each line in this file should contain a Kerberos principal name of the form `principal/instance@realm` (in a DCE cell, this principal name is referred to as `/.../cell/principal/instance`, where realm is equivalent to cell). If the originating user is authenticated to one of the principals named in `.k5login`, access is granted to the account. The principal `accountname@localrealm` is granted access if there is no `.k5login` file, and `accountname` was established as a mapping from the remote user's principal name using the Kerberos aname database. Otherwise a login and password will be prompted for on the remote machine as in `login(1)`. To avoid some security problems, the `.k5login` file should exist on the local filesystem and must be owned by the remote user.

The user must possess a ticket granting ticket which is forwardable in order to request the `-f` or `-F` options. See `kinit(1)` for information on obtaining a forwardable TGT.

If Kerberos authentication fails, a warning message is printed and the standard Berkeley rlogin is used instead.

8.2. rlogind

The user interface for the `rlogind` server, based on the BSD 4.4-Lite version, is defined below. Only the additional options related to authentication are described.

NAME

`rlogind` — remote login server

SYNOPSIS

`rlogind [-aklnxK]`

DESCRIPTION

`rlogind` is the server for the `rlogin(1)` program.

The security relevant options are as follows:

`-k`

Check authorization in `~/.k5login`. If the Kerberos authorization fails, the standard Berkeley authorization is checked (`ruserok(3M)`). If both fail, then login permission is denied.

`-K`

Similar to `-k` except that the authorization check must succeed in order for the login to succeed.

`-x`

Create an encrypted session. The `rlogind` will listen for requests on the `eklogin` port, and will decrypt all session data with the session key. If the data is not encrypted by the client with the correct key, the output will be decrypted incorrectly and will be meaningless. (This option is subject to export control.)

8.3. rsh

The user interface for the `rsh` command, based on the BSD 4.4-Lite version, is defined below. Only the additional options related to authentication are described.[4]

NAME

`rsh` — remote shell

SYNOPSIS

`rsh [-FKdfnx] [-k realm] [-l username] host [command]`

DESCRIPTION

`rsh` executes `command` on the remote host `host`.

The security relevant options are as follows:

`-k`

The `-k` option causes `rsh` to obtain tickets for the remote host in `realm` instead of the remote host's realm as determined by `krb_realmofhost(3)`.

`-x`

The `-x` option turns on DES encryption for all data exchange. This may introduce a significant delay in response time. (This option is subject to export control.)

`-K`

The `-K` option turns off all Kerberos authentication.

`-f`

The `-f` option requests `rsh` to forward the user's Kerberos credentials to the remote host, allowing the user to be authenticated without a password.

`-F`

The `-F` option causes `rsh` to forward the user's Kerberos credentials to the remote host and to allow the remote host to forward them to subsequent remote hosts.

8.4. rshd

The user interface for the `rshd` server, based on the BSD 4.4-Lite version, is defined below. Only the additional options related to authentication are described.

NAME

`rshd` — remote shell server

SYNOPSIS

`rshd [-aklnxKL]`

DESCRIPTION

`rshd` is the server for the `rsh(1)` program.

The security relevant options are as follows:

`-k`

4. The `-k` option may not be supported: see section 7.3.1.

Check authorization in ~/.k5login. If the Kerberos authorization fails, the standard Berkeley authorization is checked (ruserok(3M)). If both fail, then login permission is denied.

-K

Similar to -k except that the authorization check must succeed in order for the login to succeed.

-x

Create an encrypted session. The rshd will listen for requests on the ekshell port, and will decrypt all session data with the session key. If the data is not encrypted by the client with the correct key, the output will be decrypted incorrectly and will be meaningless. (This option is subject to export control.)

8.5. k5dcelogin

The user interface for the k5dcelogin command is defined below.

NAME

k5dcelogin — convert krb5 credentials to DCE credentials.

SYNOPSIS

k5dcelogin [cmd] [...]

(where ... means options to login(1) or to cmd)

DESCRIPTION

k5dcelogin promotes forwarded Kerberos V5 credentials to a DCE login context and executes the system login(1) command with any options specified on the k5dcelogin command line.

If cmd is specified, it is executed instead of the system login program. Any options specified are passed as arguments to the cmd.

This command is executed by the rlogind and rshd daemons as the last step of the authentication process.

8.6. dcecp

The user interface for displaying or modifying the krb5_ccache_vno and the krb5_kt_vno configuration variables is shown below.

(a) To set the credential cache version number:

dcecp -c hostvar set -krb5_ccache_vno *vno*

(b) To set the keytab version number:

dcecp -c hostvar set -krb5_kt_vno *vno*

(c) To display the credential cache version number:

dcecp -c hostvar show -krb5_ccache_vno

(d) To display the keytab version number:

dcecp -c hostvar show -krb5_kt_vno

9. API'S

9.1. sec_login_setup_identity()

The following enhancements are not committed for DCE 1.1.2, but will be implemented if there is sufficient time in the schedule.

Three new additional flags can be passed to sec_login_setup_identity() in the flags argument to request a forwardable, renewable, or proxiable ticket-granting ticket. These flags are of type sec_login_flags_t:

```
/*
 * Request a forwardable TGT.
 */
const unsigned32    sec_login_forwardable   = 0x20;

/*
 * Request a renewable TGT.
 */
const unsigned32    sec_login_renewable     = 0x40;

/*
 * Request a proxiable TGT.
 */
const unsigned32    sec_login_proxiable     = 0x80;
```

This will provide a DCE solution for requesting Kerberos V5 ticket flags without using the DCE `kinit` utility program.

10. COMPATIBILITY/INTEROPERABILITY

This section discusses the specific versions and configurations of DCE and Kerberos V5 which will be tested and supported.

Interoperability of Kerberos Version 5 clients with DCE Security Servers prior to the 1.2.2 release will not be tested and therefore is not supported.

Interoperability of Kerberos Version 5 clients based on MIT releases other than beta 4 or beta 5 will not be tested and therefore is not supported.

There is one known interoperability restriction with a DCE 1.2.2 Security Server. If the well-known DCE `pre_auth_req` Extended Registry Attribute is attached to a principal with a value of 2 (`PA-ENC-THIRD-PARTY`), the principal will not be allowed to login from a Kerberos V5 based login application. If the principal will be logging in from a Kerberos V5 client, system administrators should not attach the `pre_auth_req` ERA or attach it and set it to 0 or 1.

Credential cache and keytab file compatibility of file version formats other than version 1 or version 2 will not be tested and therefore is not supported.

Cross-realm/cell interoperability with DCE clients or a DCE Security Server prior to DCE 1.2.2 and Kerberos V5 clients or a Kerberos V5 KDC based on MIT releases other than beta 4 or beta 5 will not be tested and therefore is not supported.

REFERENCES

[Ande 95] A. Anderson, ''Hewlett-Packard White Paper: Status of DCE and Kerberos Interoperability'', Hewlett-Packard Co., March 1995.

[Brez 95] J. Brezak, ''Using MIT Kerberos5 with the DCE Security Service'', slides from OSF DCE Developers Conference, August 1995.

[John 95] G.R. Johnson, C.L. Athey, D.E. Engert, J.P. Moore, J.E. Ramus, ''Final Report and Recommendations of the Esnet Authentication Pilot Project'', Pacific Northwest Laboratory, January 1995.

[RFC 63.1] R. Salz, ''DCE 1.2 Contents Overview'', OSF RFC 63.1, May 1995.

[RFC 86.0] V. Samar, and R. Schemers ''Unified Login with Pluggable Authentication Modules'', OSF RFC 86.0, October 1995.

[RFC 1510] J. Kohl, and C. Neuman, ''The Kerberos Network Authentication Service (V5)'', IETF RFC 1510, September 1993.

AUTHOR'S ADDRESS

Sean Mullan
Hewlett-Packard Co.
300 Apollo Drive
Chelmsford, MA 01824
USA

Internet email: mullan_s@apollo.hp.com
Telephone: +1-508-436-4129
Fax: +1-508-436-4129