z/OS
2.5

Integrated Security Services Network Authentication Service Administration

SC23-6786-50

Note

Before using this information and the product it supports, read the information in "Notices" on page 225.

This edition applies to Version 2 Release 5 of z/OS® (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2022-04-25

Contents

Figures

Tables

About this document

This publication describes how to configure and administer z/OS Integrated Security Services Network Authentication Service. It supports z/OS (5650-ZOS).

Who should use this document

This document is for someone installing the product who has system programmer skills and for system administrators.

How this document is organized

This document is divided into two parts. Part 1 is the Guide; it provides overview and how-to information. Part 2 is the Reference; it contains information that you might need to look up.

The chapters in Part 1 are:

- Chapter 1, “Introducing Network Authentication Service ,” on page 3 - this chapter gives an overview of the z/OS Integrated Security Services Network Authentication Service.
- Chapter 2, “Configuring Network Authentication Service,” on page 11 - this chapter tells you how to perform the tasks done after installing the product but before using it.
- Chapter 3, “Administering Network Authentication Service ,” on page 47 - this chapter provides background information for the administrator plus the few operator commands you need.

The chapters in Part 2 are:

- Chapter 5, “Commands,” on page 73 - this chapter presents Network Authentication Service for z/OS commands in alphabetical order.
- Chapter 6, “Status codes,” on page 101 - this chapter lists the status codes for z/OS Network Authentication Service.
- Chapter 7, “Messages,” on page 155 - this chapter contains three sets of messages: messages from Kerberos runtime (EUVF02000 through EUVF03999), messages from the Security Server (EUVF04000 through EUVF05999), and messages from Kerberos commands (EUVF06000 through EUVF06999).

Where to find more information

Where necessary, this document refers to information in other documents. For complete titles and order numbers for all elements of z/OS, see *z/OS Information Roadmap*.

The companion publication for this document is *z/OS Integrated Security Services Network Authentication Service Programming*, which describes application programming interfaces for the product.

Internet sources

The softcopy z/OS publications are also available for web browsing, and PDF versions for viewing or printing from the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

You can also provide comments about this document and any other z/OS documentation by visiting that URL. Your feedback is important in helping to provide the most accurate and high-quality information.

Conventions used in this document

This document uses the following typographic conventions:

Table 1. Typographic conventions	
Font style or characters	**Explanation**
Boldface	Indicates the name of: • The item you need to select • A field, option, parameter, or command • A new term
Italic	Indicates document titles or variable information that must be replaced by an actual value.
`Monofont`	Indicates: • Names of directories, files, and user IDs • Information displayed by the system • An example • A portion of a file or sample code • A previously entered value.
Bold Monofont	Indicates information that you type into the system exactly as it appears in this document.
[ ]	Brackets enclose optional items in format and syntax descriptions.
{ }	Braces enclose a list of required items, in format and syntax descriptions, from which you must select one.
|	A vertical bar separates items in a list of choices.
< >	Angle brackets enclose the name of a key on the keyboard.
. . .	Horizontal ellipsis points indicate that you can repeat the preceding item one or more times.
\	A backslash is used as a continuation character when entering commands from the shell that exceed one line (255 characters). If the command exceeds one line, use the backslash character as the last nonblank character on the line to be continued, and continue the command on the next line.

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead "If you have a technical problem" on page xiii.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM® Documentation
If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation at ibmdoc@us.ibm.com.

Feedback on the z/OS product documentation and content
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: z/OS Integrated Security Services Network Authentication Service Administration, SC23-6786-50
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under IBM z/OS Product Documentation Update Policy (www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy?OpenDocument).

Summary of changes for z/OS Version 2 Release 5 (V2R5)

The following changes are made to z/OS Version 2 Release 5 (V2R5).

New information

The following information is new:

April 2022

- Added a range of values and a note to the description for _EUV_SVC_DBG, see “Security runtime environment variables” on page 26.
- Added a note to the DEBUG parameter of the F SKRBKDC command, see “F SKRBKDC,parameters” on page 53.
- Added new function support for Configuring KDC bind support. See the “Configuring KDC bind support” on page 26 for the configuration enhancements and “Security server environment variables” on page 32 for the two new environment variables. (APAR OA61733 also applies to V2R4).

September 2021

- Kerberos Full NDBM FIPS Support (APAR OA60507 also applies to V2R4 and V2R3) has updates in the following topics:
  - A note has been added to “Setting up a secondary KDC” on page 61.
  - “kadmin” on page 73 has been updated with a note added to the subcommands for change_password and add_policy.
- Kerberos Full NDBM FIPS Support (APAR OA60507 also applies to V2R4 and V2R3) updates the following sub commands in the kadmin command: get_principal, add_principal, modify_principal, change_password, add_key and add_policy. See “kadmin” on page 73.
- The kdb5_ndbm utility has been enhanced to handle KDC running in FIPS mode.
- The kdb5_ndbm create and stash commands have been updated with respect to the default master key encryption. See “kdb5_ndbm” on page 85.
- The kdb5_ndbm load command has been enhanced to include the following updates, see “kdb5_ndbm” on page 85:
  - A new -K keytype command option has been added.
- The keytab command has been updated to handle FIPS compliance checks. See “keytab” on page 90.

Changed information

The following information has changed.

September 2021
None

Deleted information

The following information has been deleted:

September 2021
None

New Code

- 95E73AF6

Changed Code

- None

Deleted Code

- None

New Messages

The following messages are new.

EUVF04172E
EUVF04173I
EUVF04174E
EUVF04175I
EUVF04177I
EUVF04178E
EUVF04179W
EUVF04180I
EUVF04181E
EUVF04182W
EUVF04183I
EUVF04184E
EUVF06172E
EUVF06173E
EUVF06174E
EUVF06175I

Changed Messages

The following messages are changed.

EUVF04148I (APAR OA61733)

Deleted Messages

The following messages were deleted.

None

Summary of changes for z/OS Version 2 Release 4 (V2R4)

The following changes are made to z/OS Version 2 Release 4 (V2R4).

General changes

New information

- The following topics have been updated for Flexible Authentication Secure Tunneling (FAST) support:
  - Updated the kinit command with new options -n and -T. See “kinit” on page 92.
  - A new topic has been added for Anonymous PKINIT. See “Configuration of Public Key Cryptography for initial authentication (PKINIT)” on page 14.
  - The pkinit_keyring description has been updated in the configuration profile file section for realms, see “[realms] section” on page 42.
- The following topics have been updated to include new encryption algorithms and hash algorithms:
  - The list of supported encryption types has been updated with the two new encryption types. See “Encryption types and strong encryption” on page 9.
  - The list of "for one single enctype" has been updated with the two new encryption types. See “Configuration of encryption types and FIPS level” on page 17.
  - The security runtime configuration profile has been updated to include the new checksum types and encryption types. See “Security runtime configuration profile” on page 37.
  - The libdefaults section of the configuration profile section has been updated to include the new checksum types and encryption types. See “[libdefaults] section” on page 39.
  - The sample /etc/skrb/krb5.conf configuration file has been updated to include the new default-tkt-enctypes and default-tgs-enctypes. See “Sample /etc/skrb/krb5.conf configuration file” on page 43.
  - The description of the ENCRYPT parameter has been updated in the local realm. See“Defining your local realm” on page 64.
  - The example of defining local realm has been updated. See “Example of defining the local realm” on page 64.
  - The ENCRYPT parameter in defining local principals has been updated. See “Defining local principals” on page 65.
  - The list of supported encryption types has been updated in the kadmin command. See“kadmin” on page 73.
  - The explanation of runtime codes 96C73A0E, 96C73A0F, and 96C73C30 have been updated to include the two new encryption types. See “01B79C01” on page 102.
  - The explanation of message EUVF04169E has been updated to include the two new encryption types. See “EUVF02001E” on page 155.
  - Sample Kerberos configuration KRB390.IBM.COM has been updated with the two new encryption types. See “KRB390.IBM.COM configuration” on page 211.
  - Sample Kerberos configuration KRB2003.IBM.COM has been updated with the two new encryption types. See “KRB2003.IBM.COM configuration” on page 215.

Changed information

- None

Deleted information

- None

Code Change

New Code

- 96C73A52

Changed Code

- 96C73A0E
- 96C73A0F
- 96C73C30

Deleted Code

Message changes

The following lists indicate the messages that are new, changed, or no longer issued in z/OS V2R4 and its updates. Messages that have been added, updated, or that are no longer issued in an updated edition of V2R4 are identified by the quarter and year that the message was updated, in parentheses. For example, (4Q2019) indicates that a message was updated in the fourth quarter of 2019.

New Messages

The following messages are new.

EUVF06171E

Changed Messages

The following messages are changed.

EUVF04169E
EUVF06004I

Deleted Messages

The following messages were deleted.

None

Summary of changes for z/OS Version 2 Release 3 (V2R3)

The following changes are made to z/OS Version 2 Release 3 (V2R3).

General changes

New information

- New requirements have been added for V2R3. See ./euv2b3_Overview.dita for more information.
- FIPS level information has been added to the topic for "configuration of encryption types" and the title of the topic has been renamed. See ./euv2b3_Configuration_of_encryption_types.dita.
- SKDC_FIPSLEVEL has been added to the security server environment variables table. See ./ssev.dita for more information.

Changed information

- The foreign principal definitions have been updated, see ./euv2b3_Mapping_foreign_principal_names.dita for more information.
- APAR OA53651 updates include a new optional -hkey_convert option in the kdb5_ndbm command and an update to message EUVF04085I. See the following topics for more information: ./euv2b3_kdb5_ndbm.dita and ./euvfad06.dita.
- The topic for ./enctype.dita has been updated with DES3 updates.
- The topic for ./euv2b3_Making_the_program_operational.dita has been updated to include ICSF services information.
- The foreign principal definitions have been updated, see ./euv2b3_Mapping_foreign_principal_names.dita.
- The krb5.conf sample has been updated. See ./euv2b3_Sample__etc_skrb_krb5.conf_configuration_file.dita.

Message changes

The following lists indicate the messages that are new, changed, or no longer issued in z/OS V2R4 and its updates. Messages that have been added, updated, or that are no longer issued in an updated edition of V2R4 are identified by the quarter and year that the message was updated, in parentheses. For example, (4Q2019) indicates that a message was updated in the fourth quarter of 2019.

New

The following messages are new.

96C73C2E
96C73C2F
96C73C30
96C73C31
96C73C32
96C73C33
96C73C34
96C73C35
EUVF02047W
EUVF04166I
EUVF04167E
EUVF04168E
EUVF04169E
EUVF04170E
EUVF04171E

Changed

The following messages are changed.

96C73A0E
96C73A0F
EUVF04047E
EUVF04048E
EUVF04049E
EUVF04076E
EUVF04085I for APAR OA53651

Deleted

The following messages were deleted.

None

Part 1. Guide

This part of the document contains information on:

- Introducing Network Authentication Service for z/OS
- Configuring Network Authentication for Service z/OS
- Administering Network Authentication for Service z/OS

Chapter 1. Introducing Network Authentication Service

This chapter provides an introduction to z/OS Integrated Security Services Network Authentication Service.

Overview

Integrated Security Services Network Authentication Service for z/OS is the IBM z/OS program based on Kerberos™ Version 5 and GSS. This component of z/OS will be referred to hereafter as 'Network Authentication Service forz/OS', 'z/OS Network Authentication Service', or 'z/OS Network Authentication Service'.

Network Authentication Service for z/OS provides Kerberos security services without requiring that you purchase or use a middleware product. These services include native Kerberos application programming interface (API) functions, as well as the Generic Security Service Application Programming Interface (GSS-API) functions.

Starting with z/OS version 1 release 9, the Network Authentication Service GSS-APIs support the SPKM-3/LIPKEY mechanisms. The infrastructure required to use the SPKM-3/LIPKEY mechanisms of GSS is not described in this manual, but is described in the 'Certificate/key Management' topic in *z/OS Cryptographic Services System SSL Programming*. The use of the SPKM-3/LIPKEY mechanism within GSS-APIs is documented in *z/OS Integrated Security Services Network Authentication Service Programming*.

Starting with z/OS version 2 release 2, the Network Authentication Service supports Public Key Cryptography for Initial Authentication (PKINIT).

Environmental variables that are used by the SPKM-3/LIPKEY mechanisms are GSS_KEY_LABEL, GSS_KEYRING_NAME, and GSS_KEYRING_PW. For more information on these mechanisms, see Table 6 on page 27. The GSS-API LIPKEY/SPKM mechanism codes are described in “ GSS-API LIPKEY/SPKM mechanism codes (numbers 025EA160-025EA18F)” on page 106. The RFCs that are relevant to the SPKM-3/LIPKEY mechanisms are in Table 2 on page 4.

Starting with z/OS version 2 release 3, the Network Authentication Service requires that ICSF be started and be available for the duration of operation for the KDC, Network Authentication Service commands, and Kerberos and GSS-API programs, as ICSF PKCS#11 services are used to perform all encryption and decryption functions and most hashing functions. In addition, the KDC, Network Authentication Service commands, and Kerberos and GSS-API program can be configured to run using only FIPS compliant cryptography. Along with these changes, the default encryption and checksum types have been updated to more secure types. Although ICSF is required by Network Authentication Service, there are no requirement to have cryptographic cards installed, and although PKCS#11 services are utilized, there is no requirement to configure a Token Data Set(TKDS).

Unless stated otherwise, this document mainly describes the Network Authentication Service based on the Kerberos protocol.

There is a glossary of terms for Network Authentication Service in the “” on page 229.

Supported RFCs

The following RFC numbers are supported:

Table 2. Supported RFC numbers		
RFC Area		**RFC Number**
Kerberos		• RFC 3961 (tools.ietf.org/html/rfc3961) • RFC 3962 (tools.ietf.org/html/rfc3962) • RFC 4120 (tools.ietf.org/html/rfc4120) • RFC 4537 (tools.ietf.org/html/rfc4537) • RFC 4556 (tools.ietf.org/html/rfc4556) • RFC 6113 (tools.ietf.org/html/rfc6113) • RFC 8009 (tools.ietf.org/html/rfc8009) • RFC 8062 (tools.ietf.org/html/rfc8062)
GSS		• RFC 2078 (tools.ietf.org/html/rfc2078) • RFC 2744 (tools.ietf.org/html/rfc2744)
	Kerberos Mechanism	• RFC 1964 (tools.ietf.org/html/rfc1964) • RFC 4121 (tools.ietf.org/html/rfc4121)
	LIPKEY Mechanism	• RFC 2847 (tools.ietf.org/html/rfc2847)
	SPKM-3 Mechanism	• RFC 2025 (tools.ietf.org/html/rfc2025) • RFC 2253 (tools.ietf.org/html/rfc2253) • RFC 2459 (tools.ietf.org/html/rfc2459)

Authentication

Network Authentication Service for z/OS performs authentication as a trusted third-party authentication service by using conventional shared secret-key cryptography. Network Authentication Service provides a means of verifying the identities of principals, without relying on authentication by the host operating system, without basing trust on host addresses, without requiring physical security of all the hosts on the network, and under the assumption that packets traveling along the network can be read, modified, and inserted at will.

It involves the use of a trusted third party, known as the Key Distribution Center (KDC), to negotiate shared session keys between clients and services and provide mutual authentication between them. Before the client can communicate with the server, it needs to request an initial ticket, called a ticket-granting-ticket (TGT) from the KDC and use that ticket to obtain a service ticket, which is then used for the subsequent communications with the server.

Initial-ticket exchange and Ticket-granting-ticket exchange

The initial-ticket exchange and the ticket-granting-ticket exchange, use slightly different protocols and require different API routines

The basic difference an application programmer sees is that the initial-ticket exchange does not require a ticket-granting-ticket (TGT) but does require the client to authenticate. There are different ways of initial authentication. The older way is to use the client's password. The newer way is to use the X.509 certificates. See "Configuration of Public Key Cryptography for initial authentication (PKINIT)" on page 14.

Usually, the initial-ticket exchange is for a TGT, and TGT exchanges are used from then on. In a TGT exchange, the TGT is sent as part of the request for a ticket and the reply is encrypted in the session

key that is obtained from the TGT. Thus, once a user's password is used to obtain the initial TGT, it is not required for subsequent TGT exchanges to obtain additional tickets.

A *ticket-granting ticket* contains the Kerberos server (**krbtgt/realm**) as the server name. A *service ticket* contains the application server as the server name. A ticket-granting ticket is used to obtain service tickets. To obtain a service ticket for a server in another realm, the application must first obtain a ticket-granting ticket to the Kerberos server for that realm.

The Kerberos server reply consists of a ticket and a session key, encrypted either in the user's secret key or the TGT session key. The combination of a ticket and a session key is known as a set of *credentials*. An application client can use these credentials to authenticate to the application server by sending the ticket and an *authenticator* to the server. The authenticator is encrypted in the session key of the ticket and contains the name of the client, the name of the server, and the time the authenticator was created.

To verify the authentication, the application server decrypts the ticket by using its service key, which is known only by the application server and the Kerberos server. Inside the ticket, the Kerberos server places the name of the client, the name of the server, a session key associated with the ticket, and some additional information.

The application server then uses the ticket session key to decrypt the authenticator and verifies that the information in the authenticator matches the information in the ticket. The server also verifies that the authenticator time stamp is recent to prevent replay attacks (the default is 5 minutes). Since the session key was generated randomly by the Kerberos server and delivered encrypted in the service key and a key known only by the user, the application server can be confident that users really are who they claim to be, because the user was able to encrypt the authenticator in the correct key.

To provide detection of both replay attacks and message stream modification attacks, the integrity of all the messages that are exchanged between principals can also be guaranteed by generating and transmitting a collision-proof checksum of the client’s message, keyed with the session key. The privacy and integrity of the messages exchanged between principals can be secured by encrypting the data to be passed by using the session key.

Authorization is checking if a specific user is allowed access to the requested data or resources where authentication is verifying that the user is who they say they are. Application developers must remember these are not the same thing and after authentication is complete, they still need to check to see whether the authenticated user is authorized to access the requested data or resources.

Realms

The Kerberos protocol is designed to operate across organizational boundaries. Each organization wishing to run a Kerberos server establishes its own *realm*. The name of the realm in which a client is registered is part of the client's name and can be used by the application server to decide whether to honor a request.

By establishing *inter-realm keys*, the administrators of two realms can allow a client authenticated in one realm to use its credentials in the other realm. The exchange of inter-realm keys registers the ticket-granting service of each realm as a principal in the other realm. A client is then able to obtain a ticket-granting ticket for the remote realm's ticket-granting service from its local ticket-granting service. Tickets issued to a service in the remote realm indicate that the client was authenticated from another realm.

This method can be repeated to authenticate throughout an organization across multiple realms. To build a valid authentication path to a distant realm, the local realm must share an inter-realm key with the target realm or with an intermediate realm that communicates with either the target realm or with another intermediate realm.

Realms are typically organized hierarchically. Each realm shares a key with its parent and a different key with each child. If an inter-realm key is not directly shared by two realms, the hierarchical organization allows an authentication path to be easily constructed. If a hierarchical organization is not used, it may be necessary to add a CAPATHS section to the krb5.conf configuration file in order for Kerberos to construct an authentication path between realms.

Although realms are typically hierarchical, intermediate realms may be bypassed to achieve cross-realm authentication through alternate authentication paths. It is important for the end-service to know which realms were transited when deciding how much faith to place in the authentication process. To facilitate this decision, a field in each ticket contains the names of the realms that were involved in authenticating the client.

For more information on realms, and specifically realm trust relationships, see Chapter 3, “Administering Network Authentication Service ,” on page 47.

Principals

These are the unique names of users and services used by Kerberos. They have the format Primary/Instance@Realm and will contain a password and encryption types supported by the user or service. Because the Kerberos database only contains information about users and services on the machines within the local realm, the realm name will always be the local realm except for peer trust relationships.

Note: RACF® uses a special format for entering and displaying fully qualified principal names due to the '@' being a variant character. This format is /.../Realm/Primary/Instance.

For users, the Primary is the users name, the Instance is null and the Realm is the realm assigned to the machine the user is using, for example: fred@XYZ.COM. If the user can sign on via two or more computers in the same realm then only one principal needs to be defined for that user but if the user can sign on via two or more computers in different realms then that user will need a principal defined in each realm that they can sign on from. There are some applications that will require the user to have an extra principal with an Instance related to the application, for example: fred@XYZ.COM.

For services, the Primary is the name of the service, the Instance is the machine name where the service is located and the Realm is the realm where the machine is located, for example: ftp/server01.xyz.com@XYZ.COM. Peer Trust Relationships are implemented as special Ticket Granting Service (TGS) principals where the Primary is krbtgt, the Instance is the realm name of the TGS you want to request tickets for and the Realm is the realm name of the KDC that issued the TGT. For example, krbtgt/WILMA.REALM@FRED.REALM which would be the TGT for WILMA.REALM issued by FRED.REALM).

These are the conventions but, if you are designing your own application, you can use any principal name. If you do deviate from these conventions, then you should be aware that some of the API calls might fill in missing data. For example, krb5_import_name will include the host name for a service principal if an instance is not provided, have naming restrictions (Primary and Instance cannot contain an '@' symbol) or have length restrictions (eg. RACF local principals, including the realm name, cannot exceed 240 characters).

Registry database types: SAF or NDBM

The Kerberos security server supports two registry database types: SAF (Kerberos principals stored in the System Authorization Facility database) and NDBM (Kerberos principals stored in a Unix System Services database using ZFS files). IBM recommends that the SAF registry be used unless it is necessary to share the Kerberos registry with one or more KDC instances running on another operating system.

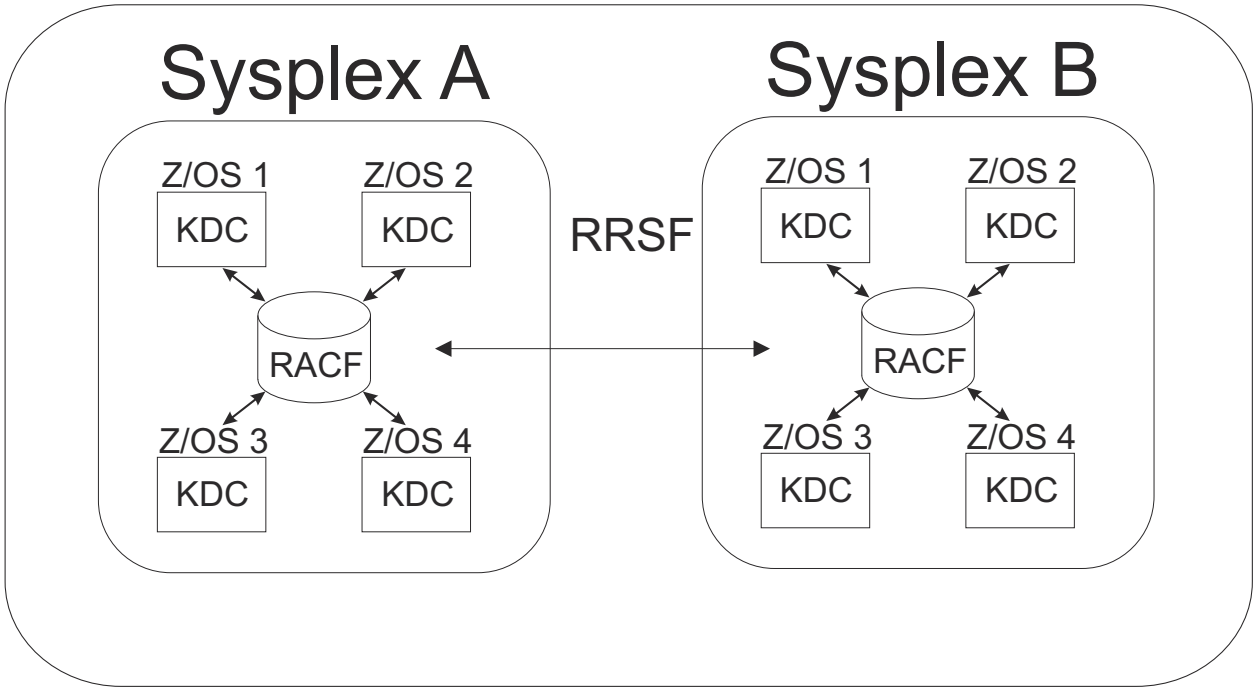

Figure 1. Cross-Sysplex Environment Using SAF Databases

The SAF registry database has these capabilities and requirements:

- Kerberos information is integrated with the z/OS system authorization profiles. All information is managed by SAF and stored in the SAF database. The KDC does not maintain its own registry database.
- The SAF database is shared within the sysplex and can be shared with other z/OS systems by using RACF Remote Sharing Facility (RRSF). Figure 1 on page 7 shows an example of this environment.
- SAF callable services are provided to map Kerberos principals to system user IDs and to map system user IDs to Kerberos principals.
- Support is provided for using system-authenticated user IDs to eliminate the use of Kerberos passwords and key tables when obtaining and decrypting tickets.
- No Kerberos administration support is provided due to semantic differences between the SAF database and the Kerberos administration wire protocols.
- All KDC instances in the realm must share the same SAF database (homogeneous environment).
- The SAF registry scales to support a large number of principals.

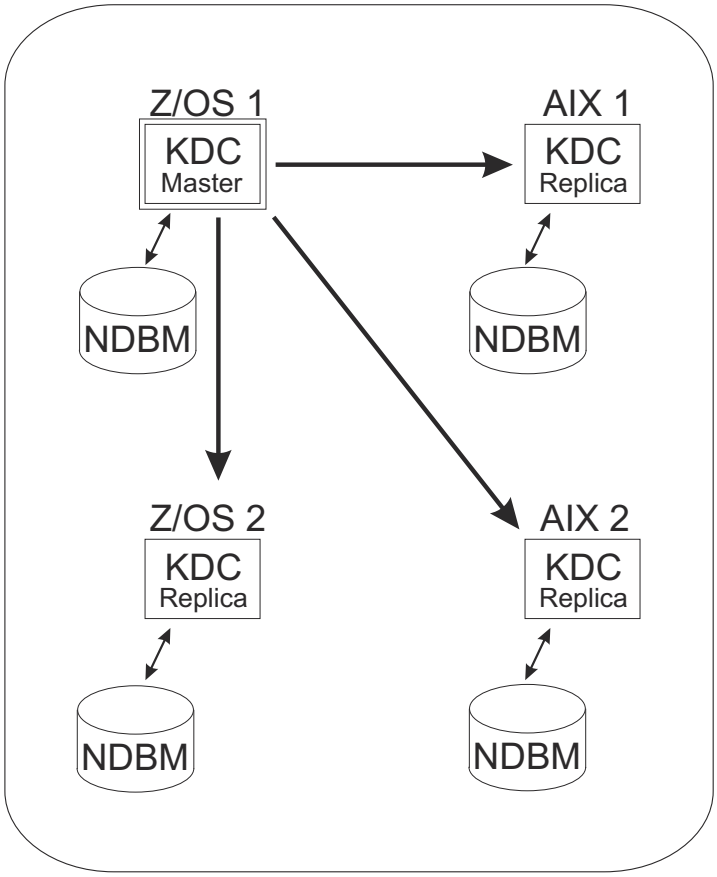

Figure 2. Cross-Platform Environment Using NDBM Databases

The NDBM registry database has these capabilities and requirements:

- The KDC maintains its own registry database using the Unix System Services NDBM support. The database files are located in the **/var/skrb/krb5kdc** directory and must be protected and backed up appropriately.
- The NDBM database is not shared within the sysplex. Each KDC instance must have its own NDBM database files. Database propagation is used to synchronize the database files for each KDC. An example of this environment is shown in Figure 2 on page 8.
- SAF callable services can be used to map Kerberos principals to system user IDs and to map system user IDs to Kerberos principals. In order to use these services, the Kerberos administrator must define the Kerberos principals in the SAF database as well as in the NDBM database. However, there is no need to synchronize the SAF password for the principal with the NDBM password for the principal since Kerberos always uses the password obtained from the NDBM database.
- System authentication can be used to eliminate the use of Kerberos passwords and key tables when obtaining and decrypting tickets. In order to use these services, the Kerberos administrator must define the Kerberos principals in the SAF database as well as in the NDBM database. However, there is no need to synchronize the SAF password for the principal with the NDBM password for the principal since Kerberos always uses the password obtained from the NDBM database.
- Full Kerberos administration support is provided.
- The realm can contain both z/OS KDC and non-z/OS KDC instances (heterogeneous environment).
- The NDBM registry is limited by the maximum size of an ZFS database. In addition, the database propagation protocol is inefficient for a large number of principals unless the update propagation protocol is used.
- The Kerberos database maintenance utility kdb5_ndbm will only operate with FIPS compliant encryption types when FIPS is enabled.

- The kadmin command used to modify the principal, attribute, and policy information can now be configured to restrict operations to only use FIPS compliant encryption keys when FIPS is enabled.

Encryption types and strong encryption

Network Authentication Service for z/OS supports the following encryption types:

- 56-bit DES, referred to specifically as DES
- 56-bit DES with key derivation, referred to specifically as DESD
- 168-bit DES, referred to specifically as DES3
- 128-bit AES, referred to specifically as AES128
- 128-bit AES SHA2 referred to specifically as AES128 SHA2
- 256-bit AES, referred to specifically as AES256
- 256-bit AES SHA2 referred to specifically as AES256 SHA2

Note: DES and DESD are only supported in non FIPS mode.

A Kerberos ticket has two portions, a user portion and a server portion, and both are encrypted with possibly different encryption types. The encryption type of the server portion is selected by the KDC as the first encryption type from the SKDC_TKT_ENCTYPES environment variable (processed left to right) that is available in the local realm definition. The encryption type of the user portion for a TGT is selected by the KDC as the first encryption type from the default_tkt_enctypes Kerberos configuration profile (processed left to right) that is available in the users principal definition. The encryption type of the user portion for a service ticket is selected by the KDC as the first encryption type from the default_tgs_enctypes Kerberos configuration profile (processed left to right) that is available in the service principal definition. The KDC does not pick encryption types based on encryption strength but on the order of the entries in the environmental variable and Kerberos configuration profile (left to right) so it is important that you make these correct as they affect the entire system. If a particular system does not support an encryption type it is not necessary to disable that encryption type for everyone but to remove it from the appropriate principal.

Although, Network Authentication Service supports DES3 and AES, due to US government export regulations, they may not be available for user data encryption. This means that tickets can be obtained for instance by using DES3 or AES encryption but the session keys in service tickets may need to be restricted to DES encryption (the session key is often used for user data encryption).

Thus, the use of DES3 or AES encryption can be controlled on an individual server basis when necessary. For example, if a foreign realm does not support DES3 or AES encryption, the **krbtgt/***foreign-realm***@***local-realm* principal entry in the KDC registry database contains just a DES key and not a DES3 or AES key.

On the other hand, if FIPS mode is enabled, only DES3 or AES can be used.

Application programming interfaces

The *z/OS Integrated Security Services Network Authentication Service Programming* is devoted to the APIs for Network Authentication Service for z/OS. This document explains how to use the APIs, as well as providing a reference section that describes each API individually.

Chapter 2. Configuring Network Authentication Service

This chapter provides configuration information for Network Authentication Service for z/OS.

Making the program operational

After you have installed z/OS, you must take certain steps to make Network Authentication Service operational. Here are those steps:

1. These steps assume that Resource Access Control Facility (RACF) is your external security manager. If you have a different but equivalent external security manager, consult the documentation for that product for the corresponding instructions and commands.

   a. Copy the SKRBKDC started task procedure from EUVF.SEUVFSAM to SYS1.PROCLIB. Change the PARM keyword to specify **–nokdc** instead of **–kdc** if you want to use the SKRBKDC application services (such as application component trace or sysplex credentials caches), but do not want to run a KDC on the system.

   b. Copy the SKRBKDC sample configuration file from **/usr/lpp/skrb/examples/krb5.conf** to **/etc/skrb/krb5.conf**.

      The file permissions should allow everyone to read the file and only the administrator to update it.

   c. Copy the SKRBKDC environment variable definitions from **/usr/lpp/skrb/examples/skrbkdc.envar** to **/etc/skrb/home/kdc/envar**. Modify the SKDC_DATABASE environment variable to select either the SAF or NDBM registry. The SAF registry is used if this environment variable is not set. IBM recommends that the SAF registry be used unless it is necessary to share the Kerberos registry with one or more KDC instances running on another operating system. The file permissions should allow only the administrator to read and update the file.

      Set the TZ value for your installation, and determine which type of database, NDBM or SAF, your site will use. TZ Is explained in the C/C++ Run-Time library reference, and instructions on modifying it are in the *z/OS XL C/C++ Programming Guide*, 'Using the TZ or _TZ Environment Variable to Specify Time Zone'.

   d. If your installation uses the SERVAUTH RACF class profiles to control access to TCP/IP ports and stacks, also use the RACF publications as a guide to update the TCP/IP resource permissions needed for Network Authentication Service users and KDCs.

   e. The IRR.RUSERMAP resource in the FACILITY class must be defined if you are are going to obtain service keys from a local instance of the KDC instead of from a key table. The application server system IDs must have RACF READ access to IRR.RUSERMAP resource to use the KRB5_SERVER_KEYTAB variable set to 1. To define IRR.RUSERMAP and grant READ authority to all system users:

      ```
      RDEFINE FACILITY IRR.RUSERMAP UACC(READ)
      SETROPTS RACLIST(FACILITY) REFRESH
      ```

      See “Security runtime environment variables” on page 26 for more on the KRB5_SERVER_KEYTAB environment variable.

   f. The following steps are to be used only if you are implementing the SAF database. Skip these steps if you are using NDBM.

      i) Before starting the SKRBKDC started task, when using the SAF database implementation, be sure that the REALM definitions and other configuration and RACF items are completed for your installation. See Chapter 3, “Administering Network Authentication Service ,” on page 47 for more information. Refer also to the appropriate sections of *z/OS Security Server RACF Security*

Administrator's Guide and supporting publications for updating the RACF Database template and the Dynamic Parsing task before using Network Authentication Service for z/OS.

ii) Define the RACF Remote Sharing Facility (RRSF) for the local system, even if you do not plan to set up an RRSF network. An RRSF local node must be defined in order to generate the corresponding Kerberos secret key whenever users change their password. Refer to *z/OS Security Server RACF Security Administrator's Guide* for information on defining the local RRSF node.

a) Create the SKRBKDC user ID with OMVS segment with a unique non zero UID, for example,

```
ADDUSER SKRBKDC DFLTGRP(SYS1) NOPASSWORD OMVS(UID(12969189)
        PROGRAM('/bin/sh') HOME('/etc/skrb/home/kdc'))
```

Note: The UID(0) requirement is lifted in z/OS V2R2. Make sure this ID is given access to the necessary directories and files.

b) Activate the APPL class if it is not already active.

```
SETROPTS CLASSACT(APPL) RACLIST(APPL)
```

iii) Define the SKRBKDC application and set the universal access to READ. Alternately, you can set the universal access to NONE and define individual groups or users to the SKRBKDC application. Users must have access to the SKRBKDC application to use the **kpasswd** Kerberos command to change their passwords.

```
RDEFINE APPL SKRBKDC UACC(READ)
```

iv) Activate the PTKTDATA class if it is not already active.

```
SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA)
```

v) Define the PassTicket data for the SKRBKDC application. PassTickets are used internally by the Kerberos security server when the user password is changed. A PassTicket is never given to a user by the Kerberos security server. The PassTicket key can be defined as either a DES key for legacy PassTickets or an HMAC key for enhanced PassTickets as described in Using PassTickets in the *z/OS Security Server RACF Security Administrator's Guide*. IBM strongly recommends the use of enhanced PassTickets.

Legacy PassTickets example:

```
RDEFINE PTKTDATA SKRBKDC UACC(NONE) SSIGNON(KEYENCRYPTED(6D9E5276BF3B1049))
```

Enhanced PassTickets example: (where SKRB.PASSTICKET.KEY is the ICSF CKDS key label)

```
RDEFINE PTKTDATA SKRBKDC UACC(NONE) SSIGNON(EPTKEYLABEL(SKRB.PASSTICKET.KEY))
```

Each Kerberos user that uses the kpasswd command to change their password will need to have UPDATE access to the IRRPTAUTH.PWCHANGE.APPL.SKRBKDC resource in the PTKTDATA class. Define the authorization profile and permit either the individual users or the group of Kerberos kpasswd users access to the resource. (this example uses a group name KRBUSERS)

```
RDEFINE PTKTDATA IRRPTAUTH.PWCHANGE.APPL.SKRBKDC UACC(NONE)
```

```
PERMIT IRRPTAUTH.PWCHANGE.APPL.SKRBKDC CLASS(PTKTDATA
     ID(KRBUSERS) ACCESS(UPDATE)
```

Alternately, you may temporarily define the profile that denies access and put it in warning mode to discover which users are using the kpasswd command, without impacting their ability to do so:

```
RDEFINE PTKTDATA IRRPTAUTH.PWCHANGE.APPL.SKRBKDC UACC(NONE) WARNING
```

vi) Refresh the APPL and PTKTDATA classes.

```
SETROPTS RACLIST(APPL PTKTDATA) REFRESH
```

g. Define the SKRBKDC started task and associate it with the SKRBKDC user ID. Define the SKRBWTR started task and associate it with the SKRBKDC user ID if you plan to perform component tracing with the SKRBWTR procedure.

```
RDEFINE STARTED SKRBKDC.** STDATA(USER(SKRBKDC))
```

```
RDEFINE STARTED SKRBWTR.** STDATA(USER(SKRBKDC))
```

h. Refresh the STARTED Class.

```
SETROPTS RACLIST(STARTED) REFRESH
```

2. If you wish, customize the **/etc/services** file to assign ports to the Kerberos services. Add the following service names and change the default entries for port/protocol to reflect how you operate the network at your installation. Each line represents a line in the **/etc/services** file showing the service name and the default values for the port/protocol. Kerberos uses the default port assignments if **/etc/services** does not contain the Kerberos entries, so customizing **/etc/services** is an optional step and only needs to be done if the default port assignments are not acceptable.

```
kerberos       88/udp
kerberos       88/tcp
kpasswd       464/udp
kpasswd       464/tcp
kerberos-adm  749/tcp
krb5_prop     754/tcp
```

3. Customize the Communications Server PROFILE DD name member to ensure that the selected ports for the KDC (usually Port 88, unless it was changed in the preceding example), the KPASSWD port (usually Port 464), and the KADMIN port (usually Port 749 unless it was changed in the preceding example) are reserved for z/OS UNIX System Services).
4. Set up the user IDs that will be using Network Authentication Service.
   a. Update the LOGON proc to add EUVF.SEUVFEXC to the SYSEXEC DD name concatenation, if the user wants to use the NAS commands from a TSO/E environment.
   b. Update the PATH environment variable to include /usr/lpp/skrb/bin in the users' UNIX System Services .profile.
5. File permissions and other considerations
   a. Configuration files in /etc/skrb
      - Make the administrator userid the owner of all the files and directories in /etc/skrb. Note: The administrator userid is assumed to be UID 0 or have the equivalent authority to read, write, and change permission and ownership of files and directories it does not own or have access. (replace xxx with the administrator userid)

        ```
        chown -R xxx /etc/skrb/
        ```

      - Change the permission for the configuration files under the /etc/skrb directory to be readable and writeable only by the administrator

        ```
        chmod -R 600 /etc/skrb
        ```

      - Change the /etc/skrb/home and /etc/skrb/home/kdc directories to allow execute permission. Also, if the KDC userid is not assigned a UID value of 0, change the owner of the directories and files within to be owned by the KDC userid. (replace yyy with the KDC userid)

        ```
        chmod 700 /etc/skrb/home /etc/skrb/home/kdc
        chown -R yyy /etc/skrb/home
        ```

      - krb5.conf (and the parent directory) need to readable by everyone.

```
chmod 644 /etc/skrb/krb5.conf
chmod 755 /etc/skrb
```

- Any keytabs in /etc/skrb need to allow the application server (not client) read access.

b. Keytab files

- Keytabs can be in any directory and should be readable only by the application servers (not clients) that need to use them and writable only by the administrator. No one else should have access.
- Keytabs should not contain keys from other realms.
- To minimize misuse of keys, split your keytabs up by application so that you have one keytab per application or one keytab per instance of an application if you run multiple instances of the same application. Why not put the keytab in the application servers home directory?
- If you are going to allow one application to use multiple service principals, then all the service principals (keys) used by that application must be in the one keytab file because the application will only be able to open the one keytab file.

c. Data files in /var/skrb

Make the started task ID the owner of this directory and its subdirectories (replace yyy with the started task ID).

```
chown yyy /var/skrb
chown yyy /var/skrb/creds
```

If NDBM is in use, and the database has already been created, make sure the directory and database files are owned by the started task userid:

```
chown -R yyy /var/skrb/krb5kdc
```

6. Since ICSF services are used for all encryption and most checksum operations, it may be necessary to permit users to have access to ICSF services used by Network Authentication Service. If the CSFSERV class is active and protection files exist for the following CSFSERV resources, the userid under which the KDC runs and all the Network Authentication Service clients and servers will need READ access to the following ICSF resources.

   When running in FIPS mode:

   CSFIQA, CSFRNG, CSFOWH, CSF1TRC, CSF1TRD, CSF1SKD, and CSF1SKE.

   When not running in FIPS mode:

   CSFIQA, CSFRNG, CSFOWH

   If the CSFSERV class is not active, or no protection profiles within the CSFSERV class are defined to restrict access to the above resources, all users on the system are permitted to use the services associated with the ICSF resources, so granting access to CSFSERV resources is not required. The authorization checks for the CSFOWH and CSFRNG resources may be disabled by defining the CSF.CSFSERV.AUTH.CSFOWH.DISABLE and CSF.CSFSERV.AUTH.CSFRNG.DISABLE.

   Network Authentication Service clients and servers include commands like kinit, and client/server programs that allow GSS-API authentication.

Configuration of Public Key Cryptography for initial authentication (PKINIT)

Public Key Cryptography for initial authentication (PKINIT) enables Network Authentication Services to run a Key Distribution Center (KDC) on z/OS to authenticate a Kerberos client from z/OS or other platforms that use the Public Key authentication method in the Authentication Service (AS) exchange. It supports two methods to encrypt the reply to the client:

- Diffie-Hellman key exchange (DH) method

- Public Key Encryption (RSA) method

To support PKINIT there are new environment variables and configuration options for both the KDC and the client to use. The KDC and the client must also set up a key ring, key token, or key database with appropriate digital certificates.

The KDC reads the environment variables set up for this mechanism in /etc/skrb/home/kdc/envar. See the options that start with SKDC_PKINIT on Table 7 on page 32 in topic "Security server environment variables" on page 32.

The client reads the krb5.conf for the configuration options in the appropriate realms section. See the options that start with pkinit in "[realms] section" on page 42.

If the keyword is repeated in the envar file or in the krb5.conf file, only the first occurrence takes effect.

The z/OS KDC requires clients to use one of two preauthentication mechanisms; PKINIT or an encrypted timestamp. The PKINIT preauthentication mechanism is the preferred mechanism when the KDC has been configured to use and support PKINIT. In addition, the KDC can be configured to require clients to use PKINIT by setting the *SKDC_PKINIT_REQUIRED*=1. Clients attempting to authenticate to the KDC without being configured to use PKINIT preauthentication but use the encrypted timestamp mechanism will fail with a preauthentication failure.When the KDC has been configured for PKINIT but does not require PKINIT preauthentication, the client may use either preauthentication mechanism. The z/OS kinit command will prefer the PKINIT preauthentication mechanism if there is sufficient client configuration to support PKINIT. If the KDC does not support PKINIT, and returns a preauthentication error, the z/OS kinit command will prompt for the password and fall back to use the encrypted timestamp mechanism.

If *SKDC_PKINIT_REQUIRED*=1, any invalid values of the SKDC_PKINIT keywords (except SKDC_PKINIT_DH_MIN_BITS) will prevent KDC from starting.

If *SKDC_PKINIT_REQUIRED*<>1, any invalid values are logged and KDC can be started. But the invalid values, except SKDC_PKINIT_DH_MIN_BITS and SKDC_PKINIT_REQUIRE_EKU, makes the KDC incapable of processing PKINIT requests.

For the z/OS client side, invalid values for pkinit_keyring, pkinit_keyring_stash in the krb5.conf file causes an error. Other invalid values cause the default values to be used.

Revocation checking

When the PKINIT preauthentication mechanism is in use, internal processing on both the KDC and client performs certificate validation. (KDC validates the clients certificate, and client validates the KDC certificate) Depending on if and how the issuing CA provides revocation information, your revocation checking configuration varies. The Network Authentication Service supports Online Certificate Status Protocol(OCSP) and Certificate Revocation Lists Distribution Points (CRL DPs).
To configure revocation checking to be performed by the KDC on client certificates, see SKDC_PKINIT_REQUIRE_REVOCATION_CHECKING on Table 7 on page 32. To configure revocation checking to be performed by the client on the KDC certificates, see pkinit_require_revocation_checking parameter for realm in the "[realms] section" on page 42.

Certificates needed in the KDC key ring, key token, or key database

- The CA certificate chains that signed the KDC certificates. If the KDC is contacted by different clients with different signing chains, multiple sets of client CA certificates are needed.
- The root CA certificate of the client chain is needed.
- The KDC certificates that meets the following requirement (There can be multiple such KDC certificates for different clients.):
  - Its private key is in the key ring / key token / key database.
  - It has usage PERSONAL (for key ring and key token).
  - If it has the keyUsage extension, it must assert the digitalSignature key usage bit.
  - The following is required by the z/OS Kerberos client:

- It must contain OtherName form in the Subject Alternative Name extension as follows:

```
OID:1.3.6.1.5.2.2 - id-pkinit-san
Value: KRB5PrincipalName ::= SEQUENCE {
realm [0] Realm,
principalName [1] PrincipalName**
}
```

  where Realm is the KDC that issues the TGS ticket, PrincipalName indicates the target realm that the ticket is for, with name type value 2 - type 2 principal name contains a sequence of a hardcoded string 'krbtgt' and the realm name that accepts the ticket.

  When the client verifies the KDC certificate, it checks whether the PrincipalName** indicated in the certificate matches the PrincipalName in the server name field in the request.

The KDC picks its certificate that was issued by the CA on the list of the trusted certifiers. It uses the first one found that satisfies the requirements.

Note: Longer processing time is experienced if a non-KDC certificate, which has a private key but has usage PERSONAL exists, since there are more potential certificates to be processed.

If key ring or key token is set up through the RACDCERT CONNECT or BIND commands, the usage can be explicitly specified by using the USAGE keyword. If key database or key token is set up through System SSL's gskkyman, the certificate usage is implicitly set – if it is added with the private key, it has usage PERSONAL.

Certificates needed in the client key ring, key token, or key database

- The CA chain certificates that signed the client certificate.
- The CA root certificates of the KDC certificates.
- The client certificate that meets the following requirements:
  - Its private key is in the key ring / key token / key database.
  - Has usage PERSONAL (for key ring and key token).
  - Marked as the default certificate.
  - If it has the keyUsage extension, it must assert the digitalSignature key usage bit.
  - Contain OtherName form in the Subject Alternative Name extension as follows:

```
OID:1.3.6.1.5.2.2  id_pkinit-san
Value: KRB5PrincipalName ::= SEQUENCE {
                     realm [0] Realm,
                        principalName [1] PrincipalName
                                                                    }
```

    where Realm is the client's realm, PrincipalName indicates the client's principal name with name type 1,

    or

```
OID: 1.3.6.1.4.1.311.20.2.3  id-ms-san-sc-logon-upn
Value: UserPrincipalName ::= UTF8STRING  (principalName@Realm)
```

  - If KDC requires extended keyusage extension (that is, the environment variable SKDC_PKINIT_REQUIRE_EKU is set to 1), the client certificate must contain,
    - One of these extended keyusage, either

      id-pkinit-KPClientauth (1.3.6.1.5.2.3.4), or
      id-ms-kp-sc-logon (1.3.6.1.4.1.311.20.2.2)
    - When an extended keyusage extension is included in the client certificate, a keyusage extension that asserts the digitalSignature key usage bit is required.

The client builds the trusted certifiers from all the other certificates in its key ring / key token / key database.

Anonymous PKINIT

When using the Flexible Authentication Secure Tunneling (FAST) support for initial authentication requests, clients will need an armor ticket to successfully use FAST. The only supported method of obtaining an armor ticket is to use anonymous PKINIT. Anonymous PKINIT is similar to PKINIT pre-authentication, however, the client does not supply a certificate or any other information that would reveal the client identity to the receiving KDC. Anonymous PKINIT requests require the same KDC configuration as PKINIT, with the exception that the root CA certificates for validating client certificates are not required. The client key rings, key tokens, or key databases are only required to contain the certificates necessary to validate the KDC certificate, as a client certificate is not used in an anonymous PKINIT authentication request. Since the anonymous PKINIT authentication request does not use a client certificate, requests will only support the Diffie_Hellman key exchange method for obtaining an anonymous tickets from the KDC, the Public Key Encryption method is not allowed.

Configuration of encryption types and FIPS level

Network Authentication Service can be enabled to run in FIPS level indicated by the 'SKDC_FIPSLEVEL' environment variable in the envar file for the KDC, and the 'fipslevel' keyword in the krb5.conf file. The value can be one of the following:

-1:FIPS mode not to be set (default, for fipslevel only)
0: non FIPS mode – this is the default for SDKC_FIPSLEVEL
1: FIPS level 1 (key strength 80 bits)
2: FIPS level 2 (key strength 112 bits, legacy use of keys can still be 80 bits)
3: FIPS level 3 (key strength 112 bits and higher, for all keys used for all operations)

Encryption types:

The **default_tkt_enctypes** value in the Kerberos configuration profile specifies the encryption types to be used for session keys in initial ticket-granting tickets. This is a list of one or more encryption types specified from most-preferred to least-preferred. The KDC selects the first supported encryption type in the list, for which it has an encryption key, when it generates the session key for an initial ticket-granting ticket. This encryption type is also used for preauthentication information.

The **default_tgs_enctypes** value in the Kerberos configuration profile specifies the encryption types to be used for session keys in service tickets. This is a list of one or more encryption types specified from most-preferred to least-preferred. The KDC selects the first supported encryption type in the list, for which it has an encryption key, when it generates the session key for a service ticket.

The three keywords/environmental variables that deal with encryption types (**default_tkt_enctypes**, **default_tgs_enctypes** and **SKDC_TKT_ENCTYPES**) process the list of encryption types from left to right until an encryption type meets the requirements needed for that environmental variable. If the list is exhausted, that is an encryption type is not selected, then an error will be reported. To specify **default_tkt_enctypes** or **default_tgs_enctypes**, specify the following in the Kerberos configuration profile (**/etc/skrb/krb5.conf**):

For one single enctype:

- specify des-cbc-crc for DES
- specify des-hmac-sha1 for DESD
- specify des3-cbc-sha1-kd for DES3
- specify aes128-cts-hmac-sha1-96 for AES128
- specify aes128-cts-hmac-sha256-128 for AES128 SHA2
- specify aes256-cts-hmac-sha1-96 for AES256
- specify aes256-cts-hmac-sha384-192 for AES256 SHA2

For multiple enctypes, specify the most preferred one first:

- specify aes256-cts-hmac-sha1-96,des-cbc-crc for AES256 and DES

Make the same updates to specify **SKDC_TKT_ENCTYPES**, located in **/etc/skrb/home/kdc/envar**.

If the keyword/variable (default_tkt_enctypes, default_tgs_enctypes or SKDC_TKT_ENCTYPES,) is not specified, the default values will be set to a list of strong algorithms, with the strongest being the first: aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1.

If SKDC_FIPSLEVEL is specified with a value greater than 0, the value specified for SKDC_TKT_ENCTYPES must have at least one strong encryption type.

If fipslevel is specified with a value greater than 0, the value specified for default_tkt_enctypes and default_tgs_enctypes must have at least one strong encryption type.

Examples:

1. Sample of /etc/skrb/home/kdc/envar FIPS level and encryption type.

   ```
   SKDC_FIPSLEVEL=2
   SKDC_TKT_ENCTYPES=des-cbc-md5,des-cbc-crc,des3-cbc-sha1
   ```

   Since FIPS level is set to 2, the first two enctypes will be rejected and des3-cbc-sha1 will be the only enctype used by the KDC.

2. Partial sample of a Kerberos configuration file libdefaults section that specifies a FIPS level and default enctypes.

   ```
   [libdefaults]
   fipslevel = 3
   default_tkt_enctypes = des-cbc-crc,aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1
   default_tgs_enctypes = aes256-cts-hmac-sha1-96,des3-cbc-sha1,des-hmac-sha1,des-cbc-md4
   ```

   Since FIPS level is set to 3, DES and DESD encryption types are disallowed and will be removed from the list when processed by the runtime. As such, the resulting list of default_tkt_enctypes will be: aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1. The resulting list of default_tgs_enctypes will be: aes256-cts-hmac-sha1-96,des3-cbc-sha1

Do not set SKDC_FIPSLEVEL or enable DES3, AES128 or AES256 ticket support until the Kerberos runtimes for all systems in the realm support encryption type DES3, AES128 or AES256. Otherwise, you can obtain tickets that cannot be processed on a given system. In addition, do not set fipslevel or enable DES3, AES128 or AES256 encryption support for user data unless all systems in the realm support these encryption types. Otherwise, you can obtain session keys that are unuseable for exchanging encrypted data.

When granting a service ticket, the KDC attempts to use the same encryption algorithm for the service ticket that was used for the ticket-granting ticket. This enables cross-realm encryption compatibility with realms that do not support the same encryption algorithms as the local realm. If the server principal does not have a key for that encryption type, then the KDC selects a key from the list specified by the SKDC_TKT_ENCTYPES environment variable.

Prior to z/OS V2R3, when the SAF database is used for principal and realm information, all supported keys are generated for a principal when the password for that principal are changed, even if some encryption types are disabled. In z/OS V2R3 and above, only the encryption keys for enabled encryption type are generated when the password is changed. When multiple z/OS releases are used, changing a principals password on an older z/OS release might not generate all the keys supported by the higher z/OS release or generate keys for encryption types that are disabled, so it is best to only change passwords on the higher z/OS release. Principals that are migrated from an older release of z/OS will not have AES keys until they change their password on a z/OS R9 or higher system. Also, if the principals were migrated from a system that had the RACF option KERBLVL set to 0, then these principals will not have DESD and DES3 keys until they change their password on a z/OS R9 or higher system.

The following table indicates the acceptable key sizes and algorithms for a given FIPS level:

Table 3. FIPS level key sizes and algorithms

Key Size or Algorithm	FIPS level 1	FIPS level 2	FIPS level 3
Minimum key size for encrypting/decrypting data	DES3 168 AES 128	DES3 168 AES 128	DES3 168 AES 128
Minimum key size for key generation	DES3 168 AES 128 DH 2048	DES3 168 AES 128 DH 2048	DES3 168 AES 128 DH 2048
Minimum key size used for signature verification	RSA 1024	RSA 1024	RSA 2048
Minimum key size used for signature generation	RSA 1024	RSA 2048	RSA 2048
Hash functions for signature generation	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512	SHA-224 SHA-256 SHA-384 SHA-512	SHA-224 SHA-256 SHA-384 SHA-512
Hash functions for signature verification	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512	SHA-224 SHA-256 SHA-384 SHA-512
Hash functions for non digital signature application	SHA-1	SHA-1 SHA-256 SHA-384	SHA-1 SHA-256 SHA-384

Security runtime configuration with LDAP and DNS considerations

1. Perform the following steps to customize **/etc/skrb/krb5.conf** for your installation:
   - Update the default_realm value with the name of your Kerberos realm
   - If you want to use LDAP to locate servers and to resolve host names, set the **use_ldap_lookup** value to 1
   - If you want to use DNS SRV and TXT records to locate servers and to resolve host names, set the **use_dns_lookup** value to 1
   - If you will not be using LDAP or DNS lookup, update the [realms] section to identify your Kerberos realm, and then update each realm that can be reached from your realm. The [realms] section is also used when a directory lookup request is unsuccessful.

     For example, if the default realm is KRB390.IBM.COM with two KDC servers using the SAF database, and a peer realm is KRB2000.IBM.COM with one KDC server, the [realms] section is defined as follows. Since the z/OS implementation combines the KDC and the password change server when the SAF database is implemented, each instance of the Kerberos security server provides both the **kdc** and the **kpasswd_server** services.

```
[realms]
KRB390.IBM.COM = {
      kdc = kdcsrvr1.krb390.ibm.com:88
      kdc = kdcsrvr2.krb390.ibm.com:88
      kpasswd_server = kdcsrv1.krb390.ibm.com:464
```

```
        kpasswd_server = kdcsrv2.krb390.ibm.com:464
    }
    KRB2000.IBM.COM = {
        kdc = winsrvr1.krb2000.ibm.com:88
    }
```

- If you will not be using LDAP or DNS lookup, update the [domain_realm] section to identify the host-to-realm mappings for your Kerberos realm and each realm that can be reached from your realm. The [domain_realm] section is also used when a directory lookup request is unsuccessful.

  Using the same example, the [domain_realm] section is defined as follows:

  ```
  [domain_realm]
  .krb390.ibm.com = KRB390.IBM.COM
  .krb2000.ibm.com = KRB2000.IBM.COM
  ```

  If no matching entry is found, the default action is to remove the first label from the host name, put what remains in uppercase, and use that for the realm name. In this case, the definitions given in the preceding examples are not really needed since they match the default.

2. If you are using DNS lookup, define the Kerberos hosts and servers in the DNS database:

   - SRV records are added for each KDC server in the realm. The Kerberos runtime searches for an SRV record using the realm name as the DNS search name. Note that DNS searches are not case-sensitive, so you cannot have two different realms whose names differ only in their case.

     The general form of the Kerberos SRV record is:

     ```
     service.protocol.realm ttl class SRV priority weight port target
     ```

     The **_kerberos** service entries define KDC instances, while the **_kerberos-adm** service entries define administration service instances, and **_kpasswd** entries define the password change service instances. Since the z/OS implementation combines the KDC and the password change server when the SAF database is implemented, each instance of the Kerberos security server provides both the KDC and the password change services. Administration service support on z/OS is available *only* when the NDBM database is implemented.

     The server entries are tried in priority order (0 is the highest priority). Server entries with the same priority are tried in a random order. **_udp** protocol records are required for the **_kerberos** and **_kpasswd** services while **_tcp** protocol records are required for the **_kerberos-adm** service. **_tcp** protocol records should be present if the server supports TCP as well as UDP requests (the z/OS Kerberos security server supports both UDP and TCP requests) for **_kerberos** and **_kpasswd**.

     For example, to define z/OS Kerberos security servers **krbsrv1** and **krbsrv2** for the KRB390.IBM.COM realm, add the following SRV records:

     ```
     _kerberos._udp.krb390.ibm.com IN SRV 0 0 88  krbsrv1.krb390.ibm.com
     _kerberos._tcp.krb390.ibm.com IN SRV 0 0 88   krbsrv1.krb390.ibm.com
     ```

     ```
     _kpasswd._udp.krb390.ibm.com  IN SRV 0 0 464 krbsrv1.krb390.ibm.com
     _kpasswd._tcp.krb390.ibm.com  IN SRV 0 0 464  krbsrv1.krb390.ibm.com
     ```

     ```
     _kerberos._udp.krb390.ibm.com IN SRV 0 0 88  krbsrv2.krb390.ibm.com
     _kerberos._tcp.krb390.ibm.com IN SRV 0 0 88  krbsrv2.krb390.ibm.com
     _kpasswd._udp.krb390.ibm.com  IN SRV 0 0 464 krbsrv2.krb390.ibm.com
     _kpasswd._tcp.krb390.ibm.com  IN SRV 0 0 464  krbsrv2.krb390.ibm.com
     ```

     An example of an **_kerberos-adm** SRV record for a z/OS primary KDC implementing the NDBM database is:

     ```
     _kerberos-adm._tcp.krb390.ibm.com IN SRV 0 0 749  krbsrv1.krb390.ibm.com
     ```

   - TXT records are added to associate host names with realm names. The Kerberos runtime searches for a TXT record starting with the host name. If no TXT record is found, the first label is removed and the search is retried with the new name. This process continues until a TXT record is found or the root is reached.

The general form of the Kerberos TXT record is:

```
service.name ttl class TXT realm
```

For example, to associate the endicott.ibm.com domain with the KRB390.IBM.COM realm, add the following TXT record:

```
_kerberos.endicott.ibm.com IN TXT KRB390.IBM.COM
```

Note that the realm name is case-sensitive in the TXT record.

3. If you are using LDAP lookup, define the Kerberos objects and attributes for the LDAP server:
   - Add a suffix definition to the LDAP server configuration file for the root domain name. For example, if your Kerberos realm names end in .COM, add the following statement to the LDAP configuration file:

     ```
     suffix "dc=COM"
     ```

   - Use the **ldapadd** command to define the default realm in the LDAP directory. Use the **/usr/lpp/skrb/examples/slapd.ldif** file as an example. Each component of the Kerberos realm name is represented by a domain component entry in the LDAP directory.

     For example, to define the KRB2000.IBM.COM realm in LDAP, use the following **ldif** definitions:

     ```
     dn: dc=COM
     dc: COM
     objectClass: domain

     dn: dc=IBM, dc=COM
     dc: IBM
     objectClass: domain

     dn: dc=KRB2000, dc=IBM, dc=COM
     dc: KRB2000
     objectClass: domain
     ```

   - Use the **ksetup** command to define the host systems in your Kerberos realm as well as the location of each security server. If you do not want to define each host system in LDAP, you can use the [domain_realm] section of the Kerberos configuration file to supply default rules to map host names to a Kerberos realm.

     Using the preceding example, define the security servers with the SAF database implementation as follows:

     ```
     addkdc kdcsrvr1.krb390.ibm.com  KRB390.IBM.COM
     addkdc kdcsrvr2.krb390.ibm.com  KRB390.IBM.COM
     addpwd kdcsrvr1.krb390.ibm.com  KRB390.IBM.COM
     addpwd kdcsrvr2.krb390.ibm.com  KRB390.IBM.COM
     addkdc winsrvr1.krb2000.ibm.com KRB2000.IBM.COM
     ```

     For the NDBM database implementation, the **addpwd** command is issued for the Primary KDC only. The **addadmin** command is also issued for the Primary KDC only. Here is an example:

     ```
     addadmin kdcsrvr1.krb390.ibm.com:749  KRB390.IBM.COM
     ```

   - Refer to *z/OS IBM Tivoli® Directory Server Administration and Use for z/OS* for more information on setting up the LDAP server.

LDAP schema definitions

If you are planning to use LDAP directory lookup functions, the Kerberos runtime requires the following LDAP schema definitions. These definitions are supplied with IBM LDAP servers as part of the IBM schema. You should also create the appropriate schema definitions for other LDAP servers.

- Integer values are represented as a signed-numeric character string with a maximum length of 11 characters
- Boolean values are represented by the character strings "TRUE" and "FALSE"

- Time values are represented as 15-byte character strings encoded in the format "YYYYMMDDhhmmssZ." All times are represented as UTC values.

Table 4. LDAP object classes

Object	Requires	Allows
domain	dc objectClass	description seeAlso
eSAP	objectClass	labeledURI sapName serviceHint
eService	objectClass	startMode startupParameters sapPtr serviceName
ibmCom1986-Krb-KerberosService	objectClass serviceName ibmCom1986-Krb-KerberosRealm	ipServicePort description seeAlso

Table 5. LDAP attributes

Attribute	Table Name	Type	Size	Access	Value
dc	dc	caseIgnoreString	64	normal	single
description	description	caseIgnoreString	1024	normal	multiple
ibmCom1986-Krb-KerberosRealm	krbRealm	caseExactString	256	normal	single
ipServicePort	ipServicePort	integer	11	normal	single
labeledURI	labeledURI	caseExactString	100	normal	multiple
sapName	sapName	caseIgnoreString	256	normal	single
sapPtr	sapPtr	DN	1000	normal	multiple
seeAlso	seeAlso	DN	1000	normal	multiple
serviceHint	serviceHint	DN	1000	normal	single
serviceName	serviceName	caseIgnoreString	256	normal	single
startMode	startMode	caseExactString	10	normal	single
startupParameters	startupParameters	caseExactString	256	normal	single

Security server configuration

Note: Since ICSF PKCS#11 or CCA callable services are used when generating the encryption keys for the realms and principals, ICSF must be running prior to configuring a security server.

Configuring the primary security server for the realm

1. Create the Kerberos definitions:

- If you plan to use SAF for the registry database, do these steps to create the Kerberos definitions in the z/OS security registry:

  a. Define the local realm and the default policy.

     For example, to define the KRB2000.IBM.COM realm with a minimum ticket lifetime of 15 seconds, a maximum ticket lifetime of 24 hours, and a default ticket lifetime of 10 hours:

     ```
     RDEFINE REALM KERBDFLT KERB(KERBNAME(KRB2000.IBM.COM)
     PASSWORD(password)
            MINTKTLFE(15) DEFTKTLFE(36000) MAXTKTLFE(86400))
     ```

  b. Create peer trust definitions.

     There will be two definitions for each trust relationship. For example, to define a trust relationship between the KRB2000.IBM.COM and WIN2000.IBM.COM realms:

     ```
     RDEFINE REALM /.../KRB2000.IBM.COM/krbtgt/WIN2000.IBM.COM
                      KERB(PASSWORD(password))
     RDEFINE REALM /.../WIN2000.IBM.COM/krbtgt/KRB2000.IBM.COM
                     KERB(PASSWORD(password))
     ```

     Note: The RDEFINE command converts the realm name to uppercase.

  c. Define the administration service. The user ID can be any acceptable name but the Kerberos principal must be **kadmin/admin**.

     ```
     ADDUSER KADMIN DFLTGRP(SYS1) PASSWORD(temporary password)
     ALTUSER KADMIN PASSWORD(password) NOEXPIRED
     KERB(KERBNAME(kadmin/admin))
     ```

  d. Define the password change server. The user ID can be any acceptable name but the Kerberos principal must be **kadmin/changepw**.

     ```
     ADDUSER CHANGEPW DFLTGRP(SYS1) PASSWORD(temporary_password)
     ALTUSER CHANGEPW PASSWORD(password)
     NOEXPIRED KERB(KERBNAME(kadmin/changepw))
     ```

  e. Add Kerberos segments to existing user definitions.

     For example, to associate the principal **test_server@KRB2000.IBM.COM** with the **krbsrv** user:

     ```
     ALTUSER KRBSRV PASSWORD(password) NOEXPIRED KERB(KERBNAME(test_server))
     ```

     Note: The ALTUSER command converts the password to uppercase if the MIXEDCASE SETROPTS option is not set. If MIXEDCASE is not set, you must ensure that the uppercase value is used when you request an initial ticket. The principal name is not converted to uppercase and the realm name is not included. You must change the password for the user in order to create the Kerberos secret key.

  f. See *z/OS Security Server RACF Command Language Reference* for more information.

- If you plan to use NDBM for the registry database instead of SAF, do these steps to create the Kerberos definitions for the NDBM database. Refer to Chapter 5, “Commands,” on page 73 for details on the Kerberos commands used in this section.

  a. Use the **kdb5_ndbm** command to create the initial registry database files. This command creates the architected principals for the Kerberos realm. It also creates two user principals, **IBMUSER** and **IBMUSER/admin**, with an initial password of IBMUSER. IBM recommends that you use the **kadmin** command and authenticate with the **IBMUSER/admin** principal to create your own administration principal after the security server is running. The passwords for the **IBMUSER** and **IBMUSER/admin** principals should then be changed or the principals should be deleted.

  b. Copy the example KDC configuration file from **/usr/lpp/skrb/examples/kdc.conf** to **/etc/skrb/home/kdc/kdc.conf**.

  c. Set the **min_life**, **max_life**, **max_renewable_life**, and **default_life** values in **kdc.conf** to the appropriate minimum ticket life, maximum ticket life, maximum renewable ticket life, and default

ticket life values. The following relationship must be observed: **min_life** <= **default_life** <= **max_life** <= **max_renewable_life**. The maximum ticket life and maximum renewable ticket life values for a specific principal are the smaller of the values specified in the principal entry and the values specified in the KDC configuration profile.

d. Set the **check_client_address** value in **kdc.conf** to 1 if you want the KDC to validate the address list contained in tickets presented by clients. Note that client address validation fails if requests pass through a firewall or a Network Address Translation (NAT) router because the client address in the ticket does not match the client address as seen by the server.

e. Add Kerberos segments to existing user definitions if you want to use SAF mapping services to map a Kerberos principal to a system user ID. These mappings are required if you plan to use Kerberos system-authentication services to eliminate the need to provide a key when requesting an initial ticket or decrypting a service ticket.

For example, to associate the principal **test_server@KRB2000.IBM.COM** with the **krbsrv** user:

```
ALTUSER KRBSRV KERB(KERBNAME(test_server))
```

2. Copy the example administration access control file from **/usr/lpp/skrb/examples/kadm5.acl** to **/etc/skrb/home/kdc/kadm5.acl** and customize it for your installation if you will be using Kerberos administration services. This file controls the Kerberos administration privileges granted to a user. Each line represents a single administration access definition, has a maximum length of 255 characters, and is assumed to be in the code page specified by the LANG environment variable. Comment lines start with a semi-colon and blank lines are ignored. Each line consists of 2 fields: the client principal name and the privileges granted. The order of the lines in the file is important because the search stops as soon as a match is found for the principal making an administration request.

The following privileges are defined. Use lowercase letters to define the granted privileges (any privilege not listed is denied) and use uppercase letters to define the denied privileges (any privilege not listed is granted). Do not mix uppercase and lowercase letters in the same definition.

Character	Privilege (granted or denied)
a	ADD is granted
A	ADD is denied
c	CHANGEPW is granted
C	CHANGEPW is denied
d	DELETE is granted
D	DELETE is denied
g	GET is granted (this may also be specified as i)
G	GET is denied (this may also be specified as I)
l	LIST is granted
L	LIST is denied
m	MODIFY is granted
M	MODIFY is denied
s	SETKEY is granted
S	SETKEY is denied
*	All privileges are granted

The client principal name can contain the following wildcards:

- ? represents a single character

- * represents zero or more characters
- Paired [ ] represent any one of the characters between the brackets.

3. Copy the example propagation control file from **/usr/lpp/skrb/examples/kpropd.acl** to **/etc/skrb/home/kdc/kpropd.acl** and customize it for your installation if you will be using Kerberos database propagation. The propagation control file contains an entry for each Kerberos security server in the realm and specifies the role assigned to each of the servers. The role should be specified as MANUAL until the secondary security server has been configured and the initial database propagation has been performed. Refer to "Kerberos database propagation" on page 59 for more information.
4. Start the SKRBKDC started task.
5. Change the passwords for **IBMUSER** and **IBMUSER/admin** if you are using the NDBM database. These principals are automatically created when the database is created and have an initial password of IBMUSER. You can use the **kpasswd** command or the **kadmin** command to change the passwords. You can also create your own administration user IDs and then delete **IBMUSER** and **IBMUSER/admin** if desired.

Configuring a secondary security server for the realm

Using a SAF registry database

The same SAF registry database must be shared by all of the security servers in the realm. This means you do not need to repeat the described RACF (or other external security manager) commands when you configure a secondary security server. Database propagation is not used by the Kerberos security server for a SAF registry database since the external security manager is responsible for any required propagation.

If the **/etc/skrb** file system is not shared between systems, copy the **/etc/skrb/krb5.conf** and **/etc/skrb/home/kdc/envar** files from the primary system to the secondary system. You do not need the **/etc/skrb/home/kdc/kadm5.acl** configuration file because Kerberos administration services are not available for the SAF registry database.

Finally, copy the SKRBKDC JCL procedure and, optionally, the message exit used to start the SKRBKDC started task. Once this has been done, you can start the SKRBKDC started task on the secondary system.

Using an NDBM registry database

The NDBM registry database is not shared by each security server in the realm (the file system containing the **/var/skrb/krb5kdc** directory must not be shared between systems). Instead, each security server maintains its own NDBM database and receives updates from the primary security server through the database propagation protocol.

If the **/etc/skrb** file system is not shared between systems, copy the **/etc/skrb/krb5.conf** and **/etc/skrb/home/kdc/envar** files from the primary system to the secondary system. Also copy the **/etc/skrb/home/kdc/kpropd.acl** configuration file. You do not need to copy the **/etc/skrb/home/kdc/kadm5.acl** configuration file since a secondary KDC does not provide Kerberos administration services for the NDBM registry database.

Use the **kdb5_ndbm** command to create the database master key stash file on the secondary system. The master key is used to decrypt Kerberos database entries. This key is not sent over the network as part of the database propagation protocol. For example:

```
kdb5_ndbm stash
```

Note: The database master password and master key encryption type must match the primary security server. It may be neccessary to specify the -k keytype option on the kdb5_ndbm stash command if the database master key encryption type of the primary security server is not the default encryption type for the current release on the secondary system.

Use the **kadmin** command to create the host key table used during database propagation. This key table is used to authenticate the secondary security server to the primary security server. The principal is

host/*system-name* where *system-name* is the host name for the secondary system. The host name must be the primary host name for the system as returned by the DNS name server. For example, to create a host principal and key table for the **dcesec4.krb390.ibm.com** system:

```
kadmin> addprinc host/dcesec4.krb390.ibm.com
kadmin> ktadd host/dcesec4.krb390.ibm.com -k /var/skrb/krb5kdc/kpropd.ktf
```

Use the **kpropd** command to receive the initial database propagation from the primary KDC for the realm. Refer to the section on database propagation for more information.

Finally, copy the SKRBKDC JCL procedure and, optionally, the message exit used to start the SKRBKDC started task. Once this has been done, you can start the SKRBKDC started task on the secondary system.

Configuring KDC bind support

The KDC is designed to query the system for all active network interfaces (IPv4 and IPv6 addresses) and bind to each active, non-loopback, non-restricted interface for inbound Kerberos requests. This is the default behavior for the z/OS KDC. The network interface queries are performed at KDC start up, and are performed at an interval determined by the SKDC_NETWORK_POLL environment variable value to detect new network interfaces or the activation of a failed network interface.

Starting with z/OS 2.4 and 2.5 with the PTFs for APAR OA61733, the z/OS KDC can be configured by the Kerberos administrator to only attempt to bind to a specified list of IP addresses. The configuration for binding to a specified list of IP addresses is enabled with new environment variables in the KDC envar file (by default, /etc/skrb/home/kdc/envar). To enable this support, the SKDC_BIND_SPECIFIED_IPADDRS_ONLY environment variable must be added to the KDC envar file and set to a value of 1. For example:

```
SKDC_BIND_SPECIFIED_IPADDRS_ONLY=1
```

For each IP address that is to be used by the KDC to bind and listen for inbound Kerberos requests, an SKDC_BIND_IPADDRn environment variable (where the "*n*" at the end of the variable name is a number from 1 to 32) is added to the KDC envar file with the value set to an IPv4 or IPv6 address in standard text format. The SKDC_BIND_IPADDRn environment variables must begin with 1 (SKDC_BIND_IPADDR1) and progress in sequence up to a maximum of 32(SKDC_BIND_IPADDR1, SKDC_BIND_IPADDR2, ... SKDC_BIND_IPADDR32). The KDC will stop processing SKDC_BIND_IPADDRn environment variables when it encounters an undefined SKDC_BIND_IPADDRn or when SKDC_BIND_IPADDR32 is processed. Other restrictions for the SKDC_BIND_IPADDRn environment variables are:

- The IPv4 and IPv6 loopback address will be ignored. (127.0.0.1 or 0:0:0:0:0:0:0:1)
- The IPv4 and IPv6 unspecified address will be ignored. (0.0.0.0 or 0:0:0:0:0:0:0:0)
- When a duplicate IP address is specified in different SKDC_BIND_IPADDRn environment variables, the duplicates will be ignored.
- When multiple occurrences of the same SKDC_BIND_IPADDRn environment variable is specified in the KDC envar file, the value of the first occurrence is used.
- If there are no valid SKDC_BIND_IPADDRn environment variables, the KDC will revert to the default behavior of binding to each active network interface.

Security runtime environment variables

Environment variables can be defined in the current command shell by using the **export** command. Environment variables can also be defined in an environment variable (**envar**) file, which is processed during security runtime initialization. Any variables that are defined through the shell override the same variables in the envar file. The _EUV_ENVAR_FILE environment variable can be used to specify the location of the envar file. By default, the **$HOME/envar** file is used.

z/OS generally sets environment variables in either /etc/profile (system-wide settings) or in $HOME/.profile (user-specific settings). Environment variables that are defined in either place override the same variables in the envar file.

The following environment variables are supported:

Table 6. Environment variables for security runtime

Environment Variable	Explanation
_EUV_ENVAR_FILE	Specifies the name of the file that contains environment variable definitions. If this variable is not set, the default is to use the envar file that is in the home directory (as specified by the _EUV_HOME or HOME environment variable). A data set name can be specified by preceding the data set name with '"//", and a DD name can be specified by preceding the DD name with"//DD:" Each line of the file consists of the variable name followed by "=" followed by the variable value with no intervening blanks or other punctuation. The variable value consists of everything following the "=" up to the end of the line (including any trailing blanks). Any line beginning with "#" is treated as a comment line. A line can be continued by ending the line with "\". The environment variables are not set until the first time that a function in the security runtime is called. Thus, it is useful for setting environment variables that are used by functions within the security runtime, although it can be used to set environment variables that are used by the application as well. In this case, the application should not rely on the environment variable values until after the security runtime is initialized. The application can access these environment variables that use the getenv() function. The environment variables are maintained by the C runtime library, so they are not available to operating system functions.
_EUV_EXC_ABEND_DUMP	Specifies whether a dump is to be generated when an abnormal termination occurs within the security runtime. This environment variable applies only to errors that are caught and processed by the security runtime. The default is to not take a dump if an abnormal termination occurs (the system can still take a dump if the exception percolates to the beginning of the condition handler stack without being handled). No dump is taken if _EUV_DUMP is set to 0 even if _EUV_EXC_ABEND_DUMP is set to enable a dump. The following values can be specified for dump control: • 0 = No dump (default) • 1 = Dump only if no CATCH/CATCH_ALL was found to handle the exception • 2 = Dump only if no explicit catch clause was found to handle the exception (that is, the exception was caught by a CATCH_ALL clause)
_EUV_HOME	The security runtime home directory is set to the value of this environment variable. If this variable is not specified, the HOME variable is used to determine the security runtime home directory. If the HOME variable is not set, the current directory is used.

Table 6. Environment variables for security runtime (continued)

Environment Variable	Explanation
_EUV_HW_CRYPTO	Specifies whether the hardware Cryptographic Support is used. A value of **0** disables the use of the hardware support, and a value of **65535** enables the use of the hardware support. The hardware support is used if this environment variable is not defined. Integrated Cryptographic Service Facility (ICSF) must be configured and running before using the hardware Cryptographic Support. Selected hardware cryptographic functions can be disabled by setting the appropriate bits to zero in the _EUV_HW_CRYPTO value. The corresponding software algorithms are used when a hardware function is disabled. For example, DES can be enabled and DES3 can be disabled by setting _EUV_HW_CRYPTO to 2. The following bit assignments are defined: • 2 = DES encryption/decryption • 4 = DES3 encryption/decryption • 8 = AES128 encryption/decryption • 16 = AES256 encryption/decryption **Note:** This environment variable has been deprecated.
_EUV_SEC_KRB5CCNAME_FILE	Specifies the name of the file that is used to locate the default Kerberos credentials cache. If this variable is not set, the default is to use the **krb5ccname** file that is in the security runtime home directory (the home directory is specified by _EUV_HOME or HOME). Precede the data set name with"//" to specify an MVS™ data set name and precede the DD name with "//DD:" to specify an MVS DD name.
_EUV_SVC_DBG	Specifies subcomponents and levels for the debug messages. Debug messages for a particular subcomponent are not logged unless the subcomponent is included in the _EUV_SVC_DBG list and the debug message level is greater than or equal to the specified level. An asterisk (*) can be used to specify all subcomponents. Debug level 1 generates the minimum amount of debug output, debug level 8 generates the maximum amount of debug output, and debug level 9 generates data dumps in addition to the debug messages. The debug level must be an integer between 0 and 9. All debug messages are suppressed for a subcomponent when its debug level is 0. The subcomponent list consists of a subcomponent name and a debug level that is separated by a period. Multiple subcomponents can be specified by separating the entries with commas. For example, `_EUV_SVC_DBG=*.1,KRB_CCACHE.8` enables debug level 1 for all subcomponents and debug level 8 for the KRB_CCACHE subcomponent. **Note:** IBM does not recommend debug tracing to be running indefinitely due to potential performance impacts.

Table 6. Environment variables for security runtime (continued)

Environment Variable	Explanation
_EUV_SVC_DBG_FILENAME	Specifies the fully qualified name of the file to receive debug messages. Debug messages are written to the file specified by the _EUV_SVC_STDOUT_FILENAME if this environment variable is not defined. If _EUV_SVC_STDOUT_FILENAME is not specified, debug messages are written to **stdout**. The current process identifier is included as part of the trace file name when the name contains a percent sign (%). For example, if _EUV_SVC_DBG_FILENAME is set to /tmp/kerberos.%.out and the current process identifier is 247, then the file name is /tmp/kerberos.247.out.
_EUV_SVC_DBG_MSG_LOGGING	Specifies whether debug messages are generated. The default is to suppress debug messages. The following values can be specified: • 0 = Suppress debug messages • 1 = Write debug messages
_EUV_SVC_DUMP	Specifies whether a dump is taken by the security runtime if a serious error is detected. This environment variable applies only to errors that are caught and processed by the security runtime. Error processing for errors that are handled by the Language Environment® (LE) runtime are controlled by the LE runtime options that are in effect at the time of the error. The following values can be specified for this value: • 0 = No dump is taken. This suppresses dumps generated as a result of an exception as well as dumps requested by the security runtime. A dump can still be taken by the operating system depending upon the nature of the error. • 1 = A dump is taken (this is the default)
_EUV_SVC_MSG_FACILITY	Specifies the facility class for messages that are written to the system logging facility. The valid facility classes are: KERN, USER, MAIL, NEWS, UUCP, DAEMON, AUTH, CRON, LPR, LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7. The default is USER.
_EUV_SVC_MSG_IDENTITY	Specifies the identity string that is prefixed to messages written to the system logging facility. The default is SKRB.
_EUV_SVC_MSG_LEVEL	Specifies the message level when logging messages. Messages that do not meet this criterion are suppressed. The default is to log all messages. The following values can be specified: • FATAL - Only fatal messages are logged • ERROR - Only fatal and error message are logged. • USER - Only fatal, error, and user messages are logged • WARNING - Only fatal, error, user, and warning messages are logged • NOTICE - Only fatal, error, user, warning, and notice messages are logged • VERBOSE - All messages are logged.

Table 6. Environment variables for security runtime (continued)

Environment Variable	Explanation
_EUV_SVC_MSG_LOGGING	Specifies the target where messages are logged. The default is to write informational messages to **stdout** and error messages to **stderr**. The following values can be specified: • NO_LOGGING = Suppress all messages • STDOUT_LOGGING = Write all messages (informational and error) to **stdout** and also write error messages to **stderr** • STDERR_LOGGING = Write informational messages to **stdout** and error messages to **stderr**. • SYSTEM_LOGGING = Write all messages to the system logging facility (**syslogd** daemon).
_EUV_SVC_STDERR_FILENAME	Specifies the fully qualified name of the file to receive standard error messages. Messages are written to **stderr** if this environment variable is not defined. The current process identifier is included as part of the file name when the name contains a percent sign (%). For example, if _EUV_SVC_STDERR_FILENAME is set to /tmp/kerberos.%.out and the current process identifier is 247, then the file name is /tmp/kerberos.247.out.
_EUV_SVC_STDOUT_FILENAME	Specifies the fully qualified name of the file to receive standard output messages. Messages are written to **stdout** if this environment variable is not defined. The current process identifier is included as part of the file name when the name contains a percent sign (%). For example, if _EUV_SVC_STDOUT_FILENAME is set to /tmp/kerberos.%.out and the current process identifier is 247, then the file name is /tmp/kerberos.247.out.
GSS_CRL_CACHE_TIMEOUT	Specifies the number of hours that a cached CRL remain valid. The valid timeout values are 0 - 720 and defaults to 24. A value of 0 disables the CRL cache.
GSS_KEY_LABEL	Specifies the label of the key that is used to authenticate the application. The default key is used if a key label is not specified.
GSS_KEYRING_NAME	Specifies the name of the key database ZFS file or the SAF key ring. A key database is used if the GSS_KEYRING_PW or GSS_KEYRING_STASH environment variable is also specified. Otherwise, a SAF key ring is used. The SAF key ring name is specified as "userid/keyring". The current user ID is used if the user ID is omitted. The user must have READ access to the IRR.DIGTCERT.LISTRING resource in the FACILITY class when using a SAF key ring owned by the user. The user must have UPDATE access to the IRR.DIGTCERT.LISTRING resource in the FACILITY class when using a SAF key ring owned by another user. Certificate private keys are not available when using a SAF key ring owned by another user.
GSS_KEYRING_PW	Specifies the password for the key database.
GSS_KEYRING_STASH	Specifies the name of the key database password stash file. The stash file name always has an extension of ".sth" and the supplied name is changed if it does not have the correct extension. The GSS_KEYRING_PW environment variable is used instead of the GSS_KEYRING_STASH environment variable if it is also specified.

Table 6. Environment variables for security runtime (continued)

Environment Variable	Explanation
GSS_LDAP_PASSWORD	Specifies the password to use when connecting to the LDAP server. This environment variable is ignored if the GSS_LDAP_USER environment variable is not defined.
GSS_LDAP_PORT	Specifies the LDAP server port. Port 389 is used if no LDAP server port is specified.
GSS_LDAP_SERVER	Specifies one or more blank-separated LDAP server host names. Each host name can contain an optional port number that is separated from the host name by a colon. The LDAP server is used to obtain CA certificates when validating a certificate and the local database does not contain the required certificate. The local database must contain the required certificates if no LDAP server is specified. Even when an LDAP server is used, root CA certificates must be found in the local database since the LDAP server is not a trusted data source. The LDAP server is also used to obtain certificate revocation lists.
GSS_LDAP_USER	Specifies the distinguished name to use when connecting to the LDAP server. An anonymous connection is used if this environment variable is not defined.
KRB5CCNAME	Specifies the default name for the credentials cache and is specified as "*type*:*name*." The supported types are FILE and MEMORY. The default credentials cache name is obtained from the credentials cache pointer file that is identified by the _EUV_SEC_KRB5CCNAME_FILE environment variable if the KRB5CCNAME environment variable is not set.
KRB5RCACHEDIR	Specifies the default replay cache directory and defaults to **/tmp**.
KRB5RCACHENAME	Specifies the default replay cache name. The Kerberos runtime generates a replay cache name if the KRB5RCACHENAME environment variable is not defined.
KRB5RCACHETYPE	Specifies the default replay cache type and defaults to **dfl**.
KRB5_CONFIG	Specifies one or more configuration file names that are separated by colons. The default configuration file is **/etc/skrb/krb5.conf**.
KRB5_KTNAME	Specifies the default key table name. The default key table name is obtained from the **default_keytab_name** configuration file entry if the KRB5_KTNAME environment variable is not defined. The default key table is **/etc/skrb/krb5.keytab** if no configuration file entry is found.

Table 6. Environment variables for security runtime (continued)	
Environment Variable	**Explanation**
KRB5_SERVER_KEYTAB	If this environment variable is set to 1, the **gss_accept_sec_context()** and **krb5_rd_req()** routines use a local instance of the Kerberos security server to decrypt service tickets instead of obtaining the key from a key table. The application must have at least READ access to the IRR.RUSERMAP facility to use this capability. The Kerberos principal that is associated with the current system identity must be the same as the Kerberos principal in the service ticket. The key table is used if the Kerberos principal for the system identity is not the same as the Kerberos principal for the service ticket. If this environment variable is set to 2, the behavior is the same as if it was set to 1 except for the following: 1. The application does not need access to the IRR.RUSERMAP facility. 2. The current system identity must have one of the following: a. Its associated principal matches the Kerberos principal in the service ticket. b. READ access in the KERBLINK class to the Kerberos principal in the service ticket. 3. A key table is not to be used even if the call to the local instance of the Kerberos security server fails.

Security server environment variables

The following environment variables are supported for the SKRBKDC started task. These variables are specified in /etc/skrb/home/kdc/envar.

Table 7. Environment variables for security server	
Environment Variable	**Explanation**
SKDC_BIND_IPADDRn	Specifies an IPv4 or IPv6 address to be bound by the KDC to listen for inbound Kerberos requests. There may be up to 32 instances of this environment variable where the "*n*" at the end of the variable name is a value from 1 to 32, and must be specified in sequence starting with SKDC_BIND_IPADDR1. This environment variable is only applicable when SKDC_BIND_SPECIFIED_IPADDRS_ONLY=1.
SKDC_BIND_SPECIFIED_IPADDRS_ONLY	Specifies whether the KDC will bind to all active network interfaces or a specified list of network interfaces for inbound Kerberos requests. **0** Disable binding to a specified list of network interfaces. (This is the default) **1** Enable support for binding to a specified list of network interfaces defined by SKDC_BIND_IPADDRn environment variables.

Table 7. Environment variables for security server (continued)

Environment Variable	Explanation
SKDC_CONSOLE_LEVEL	Specifies the message level for console logging. Kerberos security server messages are logged on the system console if the message severity is greater than or equal to the specified severity level. The valid severity levels are I, W, E, and A. The default is E if this environment variable is not defined.
SKDC_CREDS_SIZE	Specifies the credentials data space size in kilobytes, with a minimum value of 1024, a maximum value of 2097148, and a default value of 20480. The Kerberos security server stores cross-memory credentials in this data space.
SKDC_DATABASE	Specifies the type of registry database that is used by the security server: • SAF - Indicates that the security registry is maintained in the system security database available through the System Authorization Facility (SAF). The database is administered by using commands that are provided by the external security manager. The external security manager is responsible for propagating any database changes to other systems in the realm where an instance of the KDC is running. Kerberos database propagation is not used with the SAF database. • NDBM - Indicates that the security registry is maintained in ZFS files that are in the /var/skrb/krb5kdc directory. The database is administered by using Kerberos administration commands. The KDC is responsible for propagating any database changes to other systems in the realm where an instance of the KDC is running.
SKDC_FIPSLEVEL	Specifies the FIPS level which the KDC will conform. The values can be one of the following in the envar file: **0** non FIPS mode (default) **1** FIPS140-2 **2** SP800-131A with exception **3** SP800-131A without exception **Note:** The SKDC_FIPSLEVEL should match the fipslevel setting in the Kerberos configuration file used by the KDC(/etc/skrb/krb5.conf or file specified by KRB5_CONFIG environment variable).
SKDC_KADMIN_PORT	Specifies the administration service port number. If this environment variable is not defined, the administration service port is obtained from the *kerberos-adm* entry in the TCP/IP services files. If this entry is not defined, the administration service port defaults to 749. The administration service uses just the TCP protocol.

Table 7. Environment variables for security server (continued)	
Environment Variable	**Explanation**
SKDC_KPASSWD_PORT	Specifies the password change service port number. If this environment variable is not defined, the password change service port is obtained from the *kpasswd* entry in the TCP/IP services file. If this entry is not defined, the password change service port defaults to 464. The password change service uses both the UDP and TCP protocols.
SKDC_KPROP_INTERVAL	Specifies the database propagation interval in minutes and defaults to 15. The security server sends the current registry database to each secondary security server that is using the full replacement protocol. This propagation occurs at the end of each propagation interval. No propagation is done if the database is not changed since the last propagation. Secondary security servers that are using the update protocol receive database updates immediately and do not wait for the end of a propagation interval.
SKDC_KPROP_PORT	Specifies the database propagation port number. If this environment variable is not defined, the database propagation port is obtained from the *krb5_prop* entry in the TCP/IP services file. If this entry is not defined, the database propagation service port defaults to 754. Database propagation uses just the TCP protocol.
SKDC_LOCAL_THREADS	Specifies the number of threads to be used for local requests that use the S/390® Program Call instruction to communicate with the security server. The default value is 10 and the minimum value is 2.
SKDC_LOGIN_AUDIT	Specifies the wanted auditing level for login attempts (that is, granting a Kerberos initial ticket). The following values are allowed: • NONE = no auditing is done • FAILURE = only login attempts that fail due to an invalid password are audited • ALL = both success and failure login attempts are audited. The audit level is set to FAILURE if the SKDC_LOGIN_AUDIT environment variable is not specified or is set to an incorrect value. SMF type 80 records with event code 68 are written for an audit event. See *z/OS Security Server RACF Macros and Interfaces* for more information about the format of the SMF records.
SKDC_NETWORK_POLL	Specifies the network interface poll interval in minutes and defaults to 5. The security server queries the network configuration at the end of each poll interval to detect new network interfaces or the activation of a failed network interface.
SKDC_NETWORK_THREADS	Specifies the number of threads to be used for remote requests that use TCP/IP to communicate with the security server. The default value is 10 and the minimum value is 2.

Table 7. Environment variables for security server (continued)

Environment Variable	Explanation
SKDC_PORT	Specifies the KDC port number. If this environment variable is not defined, the KDC port is obtained from the *kerberos* entry in the TCP/IP services file. If this entry is not defined, the KDC port defaults to 88. The KDC uses both the UDP and the TCP protocols.
SKDC_PKINIT_REQUIRED	Specifies whether the Public Key authentication method (PKINIT) in the Authentication Service (AS) exchange is required. The following are valid values: **1** Required **0** Not required (the default)
SKDC_PKINIT_KEYRING	Specifies the key store to be used for PKINIT. The value is in the format of *<owner id>*/*<ring name>* or token name in the format of *TOKEN*/*token name*, or key database name in the format of full path key database name. For key database, a stash file also needs to be specified for the following SKDC_PKINIT_KEYRING_STASH keyword.
SKDC_PKINIT_KEYRING_STASH	Specifies the full path name of the key database stash file that contains the password of the key database. If the value of SKDC_PKINIT_KEYRING is a key database file, this entry is required.
SKDC_PKINIT_REQUIRE_EKU	Specifies whether the KDC requires the client certificate used for PKINIT to have the extended keyusage extension. The following are valid values: **1** Required **0** Not required (the default)

Table 7. Environment variables for security server (continued)

Environment Variable	Explanation
SKDC_PKINIT_REQUIRE_ REVOCATION_CHECKING	Specifies whether revocation checking is needed and what checking methods to use in PKINIT during the verification of the client certificate. Multiple checking methods can be specified separated by a comma or blanks. The valid values are as follows. The order of checking is the order that they are specified. If 'none' or any invalid value is specified in the list, no revocation is checked. **none (default)** No revocation is checked **ocsp** Revocation is checked by using the Authority Information Access (AIA) certificate extension to locate an OCSP responder to verify the certificate. **crldp** Revocation is checked by using the http format URI value in the CRLDistributionPoints extension of the certificate. **ldap** Revocation is checked by using the CRL distribution point name in the CrlDistributionPoints extension of the certificate, or the certificate issuer name on the certificate if there is no CrlDistributionPoints extension as the distinguished name of the LDAP directory entry containing the certificate revocation list (CRL) in the LDAP server. The LDAP server location is specified by the following *SKDC_PKINIT_LDAP_SERVER* keyword. If it is not specified, SKDC_PKINIT_REQUIRE_ REVOCATION_CHECKING defaults to none. **Note:** • Revocation cache values are not used. • If the revocation checking sources are provided in the certificate extensions, but cannot be contacted, the status of the certificate is considered revoked or unknown. If the revocation checking sources are not provided in the certificate extensions, the checking is skipped.
SKDC_PKINIT_LDAP_SERVER	Specifies the LDAP server location where the LDAP directory entry containing the certificate revocation list (CRL) is stored. The LDAP server is specified as: hostname[:port-number], port-number is assumed to be 389 if not specified. This is required when SKDC_PKINIT_REQUIRE_ REVOCATION_CHECKING specifies ldap.

Table 7. Environment variables for security server (continued)	
Environment Variable	**Explanation**
SKDC_PKINIT_DH_MIN_BITS	Specifies the KDC policy for the minimum Diffie-Hellman key size (in bits) to be allowed on inbound AS requests by using PKINIT. A request that uses a DH key size smaller than this value is rejected by the KDC and the KDC returns a list of supported sizes. Valid key sizes are 1024 and 2048, and defaults to 2048 if not specified or if an unsupported value is specified.
SKDC_TKT_ENCTYPES	Specifies the encryption types to be used for ticket-granting tickets and for service tickets. This is a list of one or more encryption types that are separated by commas, which are specified from most-preferred to least-preferred. When generating a ticket, the KDC selects the first entry in the list that is available for the server that is specified in the ticket. The KDC uses **des-cbc-crc** if this environment variable is not defined. Refer to “Security runtime configuration profile” on page 37for a list of available encryption types. The encryption types that are specified by the SKDC_TKT_ENCTYPES environment variable are also used by the Kerberos administration server when it generates new keys for a principal and no encryption types are specified by the administration request.

Security runtime configuration profile

The default security runtime configuration profile is **/etc/skrb/krb5.conf**. You can change this by defining the KRB5_CONFIG environment variable. You can specify multiple configuration files for the KRB5_CONFIG variable by separating the names with colons.

If a named entry can have just one value, then the first occurrence of the name is used. Otherwise, all of the entries for the same name are grouped together in the order they are encountered.

The file is divided into sections. Each section contains one or more name/value pairs with one pair per line. The name and value are separated by an equal sign. The value may be either a character string or a group of name/value pairs. If a character string is specified, it consists of all characters starting with the first non-blank character following the equal sign and continuing until the last non-blank character on the line. The maximum length of a single line in the configuration file is 2046 bytes. Comment lines are denoted by a semi-colon in the first position of the line. Blank lines are ignored.

A section name is enclosed in brackets and must appear on a line by itself. Group values are enclosed in braces with one group per line. The opening brace for a group may follow the equal sign or may be on a line by itself. The closing brace must be on a line by itself so that it won't be treated as part of the value string.

The configuration file must be in code page 1047. To support other code pages, you can use the following trigraphs:

Characters	Meaning
??(	left bracket
??)	right bracket
??<	left brace

Characters	Meaning
??>	right brace

Numeric values can be specified as follows:

Format	Meaning
ddddddd	decimal number
0ddddddd	octal number
0xddddd	hexadecimal number

Supported checksum types are:

- crc32
- rsa-md4
- rsa-md4-des
- descbc
- rsa-md5
- rsa-md5-des
- nist-sha
- hmac-sha1-des3
- hmac-sha1-96-aes128
- hmac-sha1-96-aes256
- hmac-sha256-128-aes128
- hmac-sha384-192-aes256

Supported encryption types are:

- des-cbc-crc
- des-cbc-md4
- des-cbc-md5
- des-hmac-sha1
- des3-cbc-sha1-kd
- aes128-cts-hmac-sha1-96
- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha256-128
- aes256-cts-hmac-sha384-192

Note: Previous releases of z/OS Network Authentication Service used encryption description des3-cbc-sha1. Since Network Authentication Service does not allow des3 without key derivation, that description has been changed to des3-cbc-sha1-kd. des3-cbc-sha1 will continue to be accepted, but will be converted to des3-cbc-sha1-kd.

Configuration profile file sections

The following sections of the configuration profile file are supported:

Table 8. Sections of the configuration profile file

Section	Usage
[libdefaults]	This section provides defaults for the Kerberos runtime routines.

Table 8. Sections of the configuration profile file (continued)	
Section	**Usage**
[realms]	This section defines each of the realms that can be reached from the local realm. For each realm, one or more key distribution center (KDC) hosts must be defined. The [realms] section is used if no DNS or LDAP server is available or if the desired Kerberos service is not found using the DNS or LDAP server.
[domain_realm]	This section defines the mapping between DNS names and Kerberos realm names. The [domain_realm] section is used if no DNS or LDAP server is available or if the desired mapping is not found using the DNS or LDAP server.
[capaths]	This section defines connection paths between realms. This section is not required if the Kerberos realms are arranged in a hierarchical configuration or if each realm has a peer connection to every other realm. Even in a hierarchical configuration, this section should be defined if there are direct connections between realms.

The information that follows provides details about these sections.

[libdefaults] section

ap_req_checksum_type
Specifies the default checksum type to use in an application request when a DES encryption type is in use. The default when not specified is rsa-md5. This value is ignored when the encryption type is using a DES3, AES, or DESD key.

ccache_type
Specifies the format of the credentials cache file as an integer value between 1 and 4. The default is 3.

check_delegate
Specifies whether the runtime should check the OK-AS-DELEGATE flag in service tickets. Specify 1 to check the flag and 0 to ignore the flag. If checking is enabled and the service ticket returned by the key distribution center (KDC) does not have the OK-AS-DELEGATE flag set, the **gss_init_sec_context()** function does not enable delegation for the target principal. The default is to enable checking.

clockskew
Specifies the maximum clock difference in seconds. The default is 300 (5 minutes). A Kerberos request is rejected if the difference between the server time and the request timestamp exceeds the clock skew value.

clock_offset
Specifies a fixed offset in minutes between network time and the system clock. The specified offset is added to the system clock to obtain the network time. The default is 0. The value specified by clock_offset will be overridden by the KDC time offset if kdc_timesync is set to 1.

default_keytab_name
Specifies the default key table type and name. The KRB5_KTNAME environment variable overrides this specification. The default is **/etc/skrb/krb5.keytab**.

default_realm
Specifies the default realm.

default_tgs_enctypes
Specifies one or more encryption types separated by commas and specified in most-preferred to least-preferred order. The KDC will select the first supported encryption type for the session key of service tickets. The default value if not specifies is:

- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha1-96
- des3-cbc-sha1-kd

default_tkt_enctypes
Specifies one or more encryption types separated by commas and specified in most-preferred to least-preferred order. The KDC will select the first supported encryption type for the session key of the initial ticket-granting tickets. The default value if not specifies is:

- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha1-96
- des3-cbc-sha1-kd

fipslevel
Specifies the FIPS level that commands and applications that will be required to adhere to when participating in Kerberos protocol exchanges.

-1
No FIPS level change will be attempted. This the default.

0
FIPS mode is disabled.

1
FIPS140-2

2
SP800-131A with exception

3
SP800-131A without exception

kdc_default_options
Specifies the default options used when requesting an initial ticket from the KDC as follows:

- 0x00000010 = KDC_OPT_RENEWABLE_OK
- 0x10000000 = KDC_OPT_PROXIABLE
- 0x40000000 = KDC_OPT_FORWARDABLE

Multiple options may be specified by ORing the values together. The default is 0x00000010.

kdc_req_checksum_type
Specifies the default checksum type to use in a KDC request when a DES encryption type is in use. The default when not specified is rsa-md5. This value is ignored when the encryption type is using a DES3, AES, or DESD key.

kdc_timesync
Specifies whether or not to synchronize the local time is with the KDC time. Specify 1 to synchronize the time and 0 not to synchronize the time. Do not specify 1 if the local system is running a time daemon that synchronizes the clock. The default is 0.

The time synchronization occurs when an initial ticket-granting-ticket is obtained from the KDC.

kdc_use_tcp
Set this value to 1 to use TCP stream connections instead of UDP datagrams when sending a request to the KDC. If a TCP connection cannot be established with the KDC, the runtime retries by sending a UDP datagram to the KDC. Set this value to 0 to always use UDP datagrams. The default is 0.

kpasswd_use_tcp
Set this value to 1 to use TCP stream connections instead of UDP datagrams when sending a request to the password change server. If a TCP connection cannot be established with the server, the runtime tries again by sending a UDP datagram to the password change server. Set this value to 0 to always use UDP datagrams. The default is 1.

ldap_server
Specifies the location of the LDAP server. The value consists of the host name and the port, separated by a colon. If the port is omitted, it defaults to 389.

rsa_md4_des_compat
Beta versions of Kerberos Version 5 computed the checksum incorrectly for the rsa-md4-des checksum type. Specify 1 to use the old algorithm for compatibility with these beta versions of Kerberos Version 5. The default is to use the new algorithm.

rsa_md5_des_compat
Beta versions of Kerberos Version 5 computed the checksum incorrectly for the rsa-md5-des checksum type. Specify 1 to use the old algorithm for compatibility with these beta versions of Kerberos Version 5. The default is to use the new algorithm.

safe_checksum_type
Specifies the default checksum type for a safe request. The default is **rsa-md5-des**. The specified checksum type must be compatible with the session key encryption type if the checksum uses an encrypted hash. When a DES3 or AES encryption key is used, this value is ignored. The following shows the checksum types that use an encrypted hash and the required session key encryption type:

Checksum Type	Encryption Type
descbc	des-cbc-crc
rsa-md4-des	des-cbc-md4
rsa-md5-des	des-cbc-md5
hmac-sha1-des3	des3-cbc-sha1
hmac-sha1-96-aes128	aes128-cts-hmac-sha1-96
hmac-sha1-96-aes256	aes256-ctc-hmac-sha1-96
hmac-sha256-128-aes128	aes128-cts-hmac-sha256-128
hmac-sha384-192-aes256	aes256-cts-hmac-sha384-192

use_dns_lookup
Set this value to 1 to use the domain name service (DNS) name server to locate the KDC and to resolve host names. The KDC is located using SRV records, and host names are resolved to realm names using TXT records. The [realms] and [domain_realm] sections are used if the resolution is unsuccessful using the DNS name server. Set this value to 0 to bypass the DNS lookup step. The default is 0. The priority value for SRV records is used to order the service records. Entries with the same priority are randomly selected each time the client needs to contact a Kerberos server.

use_dvipa_override
Set this value to 1 to allow the principal in the incoming service ticket to override the principal specified on the **krb5_rd_req**, **krb5_rd_req_verify** or **gss_accept_sec_context** API call provided only the instance (host name) of the two principals is different. If the primary or realm portion of both principals are different or either principal is a nonstandard service principal (does not have an instance or has 2 or more instances) then the incoming ticket is rejected. The application will require access to the encryption keys for the principal in the incoming service ticket (either via a keytab file or via the KDC if KRB5_SERVER_KEYTAB is set) to decrypt the ticket. Ensure the Kerberos server is running on all system images where the application runs when the value is set to 1. Set this value to 0 to only allow incoming service tickets that match the service principal specified on the **krb5_rd_req**, **krb5_rd_req_verify** or **gss_accept_sec_context** API call. The default is 0.

use_ldap_lookup
Set this value to 1 to use the Lightweight Directory Access Protocol (LDAP) directory to locate the KDC and to resolve host names. The [realms] and [domain_realm] sections are used if the resolution is unsuccessful. Set this value to 0 to bypass the LDAP lookup step. The default is 0. If both LDAP and DNS are used, LDAP is checked first, followed by DNS. The *ldap_server* value must also be specified to use LDAP lookup. LDAP directory entries are randomly selected each time the client needs to contact a Kerberos server.

[realms] section

The realms section contains one or more entries of realm value = (group definition). The realm value is a Kerberos realm name. The value is a group definition that defines the Kerberos servers for the realm. Each realm that can be contacted by applications on the local system must have an entry in the [realms] section of the configuration file unless DNS or LDAP lookup is enabled. The group entry consists of one or more occurrences of the following keywords.

kdc
The value for each kdc name entry is the host name, the port that is assigned to the kdc on that system, and the protocol (UDP or TCP), separated by colons. If the port is omitted, it defaults to 88. If the protocol is omitted, the entry can be used with both protocols.

admin_server
The value for each admin_server entry is the host name and the port that is assigned to the administration service on that system, which is separated by a colon. If the port is omitted, it defaults to 749. The protocol is always TCP for the administration service.

kpasswd_server
The value for each kpasswd_server entry is the host name, the port assigned to the password service on that system, and the protocol (UDP or TCP), separated by colons. If the port is omitted, it defaults to 464. If the protocol is omitted, the entry can be used with both protocols.

pkinit_keyring
This keyword specifies the client's RACF key ring name value in the format of <owner id>/<ring name> or token name in the format of *TOKEN*/<token name>, or key database name in the format of <full path key database name> together with the stash file name. If usage of this key store is only for anonymous PKINIT support, only certificates required for validating KDC certificates are required to be in the key ring, key token, or key database.

pkinit_keyring_stash
The pkinit_keyring_stash keyword specifies the stash file name in the format of <full path stash file name> containing the password of the key database. If value of the pkinit_keyring is a key database file, this entry is required.

pkinit_require_revocation_checking
The pkinit_require_revocation_checking keyword indicates whether revocation checking is needed and what checking methods are used during the verification of the certificate. The valid values are:

none
No revocation is checked (default).

ocsp
Revocation is checked by using the Authority Information Access (AIA) certificate extension to locate an OCSP responder to verify the certificate. web server must be running in order for this checking to be successful.

crldp
Revocation is checked by using the HTTP format URI values in the CrlDistributionPoints extension of the certificate.

ldap
Revocation is checked by using the CRL distribution point name in the CrlDistributionPoints extension of the certificate, or the certificate issuer name on the certificate if there is no CrlDistributionPoints extension as the distinguished name of the LDAP directory entry containing the certificate revocation list (CRL) in the LDAP server.

The LDAP server location is specified by the following pkinit_ldap_server keyword. If it is not specified, pkinit_require_revocation_checking will default to 'none'.

Note:

- Revocation cache values are not used.

- If the revocation checking sources are provided in the certificate extensions but cannot be contacted, the status of the certificate is considered revoked or unknown. If the revocation checking sources are not provided in the certificate extensions, the checking is skipped.

pkinit_ldap_server
The pkinit_ldap_server keyword indicates the LDAP server location for checking certificate revocation status. This is required when pkinit_require_revocation_checking specifies ldap.

pkinit_rsa_protocol
The pkinit_rsa_protocol keyword indicates if the request uses the Diffie-Hellman exchange method or the RSA public key encryption to encrypt the reply. The absence of this keyword or any values other than 1 defaults to 0. If the Kerberos configuration indicates the use of the RSA protocol and the -n option is specified on the kinit command, the RSA protocol is disabled and the Diffie-Hellman key agreement is used.

0 use DH
1 use RSA

pkinit_dh_min_bits
The pkinit_dh_min_bits keyword indicates the minimum key size, in bits, of the Diffie-Hellman key to be generated. Valid values are 1024 or 2048, and defaults to 2048 if not specified or if an unsupported value is specified. It is ignored if pkinit_rsa_protocol is set to 1.

pkinit_kdc_hostname
The pkinit_kdc_hostname keyword indicates whether the client is willing to accept a KDC certificate with a dNSName SAN (Subject Alternative Name) rather than requiring the id-pkinit-san as defined in RFC 4556. Its value should contain the acceptable host name for the KDC (as contained in its certificate). If this keyword is repeated for different values, the values are searched for matching until a match is found.

[domain realm] section

hostname
The *hostname* value is a DNS host name. The value is the name of the Kerberos realm that contains the specified host system.

.suffix
The *.suffix* value is the domain portion of a DNS host name. The value is the name of the Kerberos realm that contains host systems in the specified domain. A specific host name definition takes precedence over the domain specification.

If a matching entry is not found for a particular host name, the default is to remove the first label, put what remains in uppercase, and use that for the realm name. For example, if no match is found for **host25.krb390.ibm.com**, the realm name is set to KRB390.IBM.COM.

[capaths] section

realm
Each *realm* value is a Kerberos realm name and represents the starting point for a request. If the configuration file is not shared between systems, then the only realm that needs to be specified is the local realm. Otherwise, there needs to be a realm definition for each system sharing the configuration file. The value is a group definition that defines the target realms. If multiple trust hops are required to reach the target realm, there are multiple entries defining each of the trust relationships from the local realm to the target realm. If there is a trust relationship between the local realm and the target realm, specify the hop as a period.

Sample /etc/skrb/krb5.conf configuration file

```
[libdefaults]

default_realm = KRB390.IBM.COM
kdc_default_options = 0x40000010
use_dns_lookup = 0
```

```
; Specify the FIPS level
; -1: not to be set, use the current level of the running program this is the default
;  0: non FIPS mode
;  1: FIPS level 1 (key strength 80 bits)
;  2: FIPS level 2 (key strength 112 bits, legacy use of keys can still be 80 bits)
;  3: FIPS level 3 (key strength 112 bits and higher, for all keys used for all operations)
;fipslevel = -1
; Default encryption types
default_tkt_enctypes=aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1
default_tgs_enctypes=aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1
; Enable DES encryption types (AES, DES3 and DESD are disabled)
;default_tkt_enctypes = des-cbc-md5,des-cbc-md4,des-cbc-crc
;default_tgs_enctypes = des-cbc-md5,des-cbc-md4,des-cbc-crc
; Enable DES3 and DES encryption types (AES and DESD are disabled)
;default_tkt_enctypes = des3-cbc-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc
;default_tgs_enctypes = des3-cbc-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc
; Enable all encryption types
;default_tkt_enctypes = aes256-cts-hmac-sha384-192,aes128-cts-hmac-sha256-128,aes256-cts-hmac-
sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1,des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc
;default_tgs_enctypes = aes256-cts-hmac-sha384-192,aes128-cts-hmac-sha256-128,aes256-cts-hmac-
sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1,des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc
; Enable FIPS compliant encryption types
;default_tkt_enctypes = aes256-cts-hmac-sha384-192,aes128-cts-hmac-sha256-128,aes256-cts-hmac-
sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1
;default_tgs_enctypes = aes256-cts-hmac-sha384-192,aes128-cts-hmac-sha256-128,aes256-cts-hmac-
sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1
```

```
[realms]

KRB390.IBM.COM = {
    kdc = dcesec4.endicott.ibm.com:88
    kpasswd_server = dcesec4.endicott.ibm.com:464
    admin_server = dcesec4.endicott.ibm.com:749
    kdc = dcesec7.endicott.ibm.com:88
    kpasswd_server = dcesec7.endicott.ibm.com:464
    admin_server = dcesec7.endicott.ibm.com:749

;   RACF key ring name value in the format of <owner id>/<ring name>:
;     pkinit_keyring = KRBUSR/KRBRING
;   or token name in the format of *TOKEN*/<token name>:
;     pkinit_keyring = *TOKEN*/KRBTOKEN
;   or key database name in the format of <full path key database name>
;   together with the stash file name in the format of
;   <full path stash file name> containing the password of the key
;   database:
;     pkinit_keyring = /etc/skrb/clientgsk.kdb
;     pkinit_keyring_stash = /etc/skrb/clientgsk.sth

;   Indicate whether revocation checking is needed and what checking
;   method(s) should be used during the verification of the
;   certificate.
;   The valid values are:
;    none  - No revocation will be checked (default)
;    ocsp  - Revocation is checked using the Authority Information
;            Access (AIA) certificate extension to locate an OCSP
;            responder to verify the certificate.
;    crldp - Revocation is checked using the HTTP format URI value(s)
;            in the CrlDistributionPoints extension of the certificate.
;    ldap  - Revocation is checked using the CRL distribution point
;            name in the CrlDistributionPoints extension of the
;            certificate, or the certificate issuer name on the
;            certificate if there is no CrlDistributionPoints extension
;            as the distinguished name of the LDAP directory entry
;            containing the certificate revocation list (CRL) in the
;            LDAP server.
;   Multiple checking methods can be specified separated by a comma or
;   blank(s). The order of checking is the order they are specified.
;   For example, try ocsp first, then crldp, then ldap:
;     pkinit_require_revocation_checking = ocsp,crldp,ldap
;
    pkinit_require_revocation_checking = none

;   LDAP server name in the form of host name is followed by an
;   optional port number separated by a colon.  Ignored if ldap is
;   not specified above.
;
;     pkinit_ldap_server = myldap.server.com:389
```

```
;   Request to use the Diffie-Hellman exchange method or the RSA
;   public key encryption to encrypt the reply. The absence of this
;   keyword defaults to 0.
;     0 - use DH
;     1 - use RSA
;
    pkinit_rsa_protocol = 0

;   The minimum key size, in bits, of the Diffie-Hellman key to be generated.
;   If this keyword is absent, the default value is 2048. Ignored if
;   pkinit_rsa_protocol is set to 1.
;
    pkinit_dh_min_bits = 2048

;   This option indicates that the client is willing to accept a KDC
;   certificate with a dNSName SAN (Subject Alternative Name) rather
;   than requiring the id-pkinit-san as defined in RFC 4556.
;   Its value should contain the acceptable hostname for the KDC (as
;   contained in its certificate).
;   This option may be specified multiple times.
;
;     pkinit_kdc_hostname = alps1999.pok.ibm.com

}

KRB2000.IBM.COM = {
    kdc = sstone1.krb2000.ibm.com:88
    kpasswd_server = sstone1.krb2000.ibm.com:464
}

[domain_realm]

.krb2000.ibm.com = KRB2000.IBM.COM
.endicott.ibm.com = KRB390.IBM.COM
```

Chapter 3. Administering Network Authentication Service

This chapter provides information on administering Network Authentication Service for z/OS.

Adding principals

The following details how to add principals to the database.

Local principals

Local principals can be added to an NDBM database using the kadmin command (Refer to the description of the kadmin command for more information). For a RACF database, the Kerbname value of the altuser command can be used to associate a Kerberos principal with an existing RACF user. The kerbname value of the RACF adduser command can be used to create a new user id and associate it with a Kerberos principal (refer to *z/OS Security Server RACF Command Language Reference* for the format of these commands).

Foreign principals

Some applications require that you associate principals with local RACF identities. For Foreign Kerberos principals, this can be done by using the KERBLINK class (refer to Mapping Foreign principals in *z/OS Security Server RACF Security Administrator's Guide*). Note that the association of a foreign Kerberos principal with a local RACF Identity via the KERBLINK class does not create a local Kerberos segment, as is done by the adduser and altuser commands. RACF services such as R_USERMAP can only be used to map the RACF ID to the local kerberos segment, but can map many foreign principals to the local RACF ID using the KERBLINK association and the Kerberos segment. Creating a KERBLINK association is not required for local principals added by the adduser and altuser commands.

Principal names

All principal names and passwords should consist only of characters from the POSIX portable character set, but should not include any variant characters, such as brace, bracket, or currency symbols. See *z/OS Integrated Security Services Network Authentication Service Programming* for a table of the POSIX characters. In addition, principal names should not contain blanks or the commercial "at" sign (@).

If for some reason you need to use blanks in a principal name as part of a command-line argument, enclose the whole name in quotation marks.

If you have a need to use principal names or passwords that contain characters not in the POSIX portable character set (in other words, for national language reasons), be sure that the LANG value for the SKRBKDC started task is set to a code page that translates the national language characters to those that RACF can use. This applies to z/OS clients as well.

For users, principal names can be chosen to match their user ids. For services the format of the name to be defined in the KDC and/or in the keytab file has the format service_name/realm_name.

For services, the service_name depends on the Kerberized application, and could use the word "host" or the name of the application. For example, for Kerberized ftp the principal name at realm dcesec4.krb390.ibm.com could be ftp/dcesec4.krb390.ibm.com or host/dcesec4.krb390.ibm.com. The service principal name in the KDC or the keytab must match the name used by the Kerberized application - please consult the documentation of your Kerberized application.

Realm trust relationships

Network Authentication Service for z/OS supports two types of trust relationships: peer and transitive.

Peer trust

In a peer trust relationship, two realms exchange secret keys so that one realm can create a ticket-granting ticket (TGT) that will be accepted by the other realm. The trust relationship is established by defining a pair of principals in each realm. For example, if a peer trust relationship is to be established between KRB390.IBM.COM and KRB2000.IBM.COM, the following principals must be defined in each realm:

```
krbtgt/KRB390.IBM.COM@KRB2000.IBM.COM
krbtgt/KRB2000.IBM.COM@KRB390.IBM.COM
```

Principal names beginning with **krbtgt/** are reserved for this purpose and must not be used for other purposes.

Transitive trust

In a transitive trust relationship, two realms trust each other if they trust the intermediate realms involved in granting a ticket. Kerberos transitive trust is based upon a hierarchical trust path between the ticket client and the ticket server. The components in the trust path are formed by using periods to separate the realm name into its constituent parts. The common portion between the two realm names forms the top ancestor in the trust path.

For example, if the client is in realm SSTONE1.KRB2000.IBM.COM and the server is in realm DCESEC4.KRB390.IBM.COM, the trust path consists of the following entries:

- SSTONE1.KRB2000.IBM.COM
- KRB2000.IBM.COM
- IBM.COM
- KRB390.IBM.COM
- DCESEC4.KRB390.IBM.COM

If each realm involved in granting the service ticket is present in the trust path, then the ticket is trusted.

When attempting to obtain a service ticket, the Kerberos runtime starts with the client realm and attempts to obtain a TGT for the server realm. If the KDC for the client realm is unable to satisfy the request because there is no peer trust relationship between the client and server realms, the Kerberos runtime attempts to obtain a TGT to a realm that is in the trust path between the client and the server realms. As soon as it obtains a TGT to an intermediate realm, it tries to obtain a TGT to the server realm from the intermediate KDC. This process is repeated until either a server realm TGT is obtained or all of the intermediate realms have been tried.

When setting up transitive trust, a peer trust relationship should be defined between each realm and a common ancestor realm in the hierarchy. For example, consider the following realm tree:

- IBM.COM (Top node)
- KRB390.IBM.COM (First level node)
- DCESEC4.KRB390.IBM.COM (Second level node)
- DCESEC7.KRB390.IBM.COM (Second level node)
- KRB2000.IBM.COM (First level node)
- SSTONE1.KRB2000.IBM.COM (Second level node)
- SSTONE2.KRB2000.IBM.COM (Second level node)

A fully-connected hierarchy has, at a minimum, the following peer trust relationships:

- DCESEC4.KRB390.IBM.COM <--> KRB390.IBM.COM
- DCESEC7.KRB390.IBM.COM <--> KRB390.IBM.COM
- KRB390.IBM.COM <--> IBM.COM
- KRB2000.IBM.COM <--> IBM.COM

- SSTONE1.KRB2000.IBM.COM <--> KRB2000.IBM.COM
- SSTONE2.KRB2000.IBM.COM <--> KRB2000.IBM.COM

An additional peer trust relationship can be defined to shorten the transited path between the lower layer of realms:

- KRB390.IBM.COM <--> KRB2000.IBM.COM

Do not include the top node in the trust hierarchy if there is no need to obtain tickets for that realm. In the preceding example, IBM.COM could be omitted and the peer trust relationships would then be the following:

- DCESEC4.KRB390.IBM.COM <--> KRB390.IBM.COM
- DCESEC7.KRB390.IBM.COM <--> KRB390.IBM.COM
- KRB390.IBM.COM <--> KRB2000.IBM.COM
- SSTONE1.KRB2000.IBM.COM <--> KRB2000.IBM.COM
- SSTONE2.KRB2000.IBM.COM <--> KRB2000.IBM.COM

Similarly, a lopsided trust hierarchy can be defined. Suppose there is no need to obtain tickets to the IBM.COM or KRB2000.IBM.COM realms. The peer trust relationships would then be the following:

- DCESEC4.KRB390.IBM.COM <--> KRB390.IBM.COM
- DCESEC7.KRB390.IBM.COM <--> KRB390.IBM.COM
- SSTONE1.KRB2000.IBM.COM <--> KRB390.IBM.COM
- SSTONE2.KRB2000.IBM.COM <--> KRB390.IBM.COM

When defining the transitive trust hierarchy, it is important to remember that the peer trust relationships must be symmetric (tickets can be obtained when traversing the trust path in either direction) and each realm in a peer trust relationship must be capable of either providing a TGT for the destination realm or for an intermediate realm which is further along the trust path between the client and the server realms.

Passwords

The **krbtgt** principals are used for ticket-granting tickets. When using the SAF database, KERBDFLT is used for your local realm (**krbtgt**/*local-realm*@*local-realm*) and the RDEFINE global name (**/.../***realm-name***/***principal-name*) is used for peer-to-peer connections. In this respect, each pair of peer-to-peer principals is repeated in the foreign registries and must have the same passwords.

For example, if you are connecting REALMA and REALMB using the SAF database implementation, you would have the following RDEFINE statements in *both* REALMA and REALMB:

```
RDEFINE REALM /.../REALMA/KRBTGT/REALMB KERB(PASSWORD(PSWD1))
RDEFINE REALM /.../REALMB/KRBTGT/REALMA KERB(PASSWORD(PSWD2))
```

For the NDBM database, the **krbtgt**/REALMB@REALMA principal is used by the REALMA KDC to grant a ticket to REALMB. Similarly, the **krbtgt**/REALMA@REALMB principal is used by the REALMB KDC to grant a ticket to REALMA. These principals must be added to the NDBM database by the **kadmin** command to establish peer trust.

A password can be any value as long as you specify the same password each time you define the principal.

Cache files

The Kerberos runtime stores network credentials in cache files located in **/var/skrb/creds**. These files should be erased periodically. There are several ways to do this:

- Use a temporary file system mounted at **/var/skrb/creds**. This results in all the credentials cache files being deleted each time the system is restarted.

- Erase all of the files in **/var/skrb/creds** when the **/etc/rc** initialization script is run. This results in all of the credentials cache files being deleted each time the system is restarted.
- Set up a **cron** job to run the **kdestroy** command with the **-e** option. This results in the deletion of only expired credentials cache files. This is the preferred method for managing the credentials cache files. The **cron** job should run with UID 0 so that it can delete the cache files.

Audit

SMF Type 80 records are created for login requests (Kerberos initial ticket requests). Both success and failure events can be logged as determined by the SKDC_LOGIN_AUDIT environment variable. The event code is 68 and the record includes relocate sections 333 (Kerberos principal name), 334 (request source), and 335 (KDC error code).

The Kerberos principal is stored as a global name (**/.../realm-name/principal-name**) and not as a Kerberos name (**principal-name@realm-name**). This is done to avoid code page problems caused by the at-sign variant character. If the request is received through TCP/IP, the request source is the network address (*nnn.nnn.nnn.nnn:ppppp*). If the request is received through Program Call, the request source is the system user ID of the requester The KDC error code is a value between 0 and 127.

KDC error codes

The possible KDC error codes are:

0
No error

1
Client entry is expired

2
Server entry is expired

3
Protocol version is not supported

4
Client key is encrypted in an old master key

5
Server key is encrypted in an old master key

6
Client is not defined in the security registry

7
Server is not defined in the security registry

8
Principal is not unique in the security registry

9
No key is available for the principal

10
Ticket is not eligible for postdating

11
Ticket is never valid

12
Request rejected due to KDC policy

13
Request option is not supported

14
Encryption type is not supported

15
Checksum type is not supported

16
Preauthentication type is not supported

17
Transited data type is not supported

18
Client account is revoked

19
Server account is revoked

20
TGT is revoked

21
Client account is not valid yet

22
Server account is not valid yet

23
Password is expired

24
Preauthentication failed

25
Preauthentication required

26
Supplied authentication ticket is not for the requested server

27
Server requires user-to-user protocol

31
Decryption integrity check failed

32
Ticket is expired

33
Ticket is not valid yet

34
Request is a replay of a previous request

35
Supplied authentication ticket is not for the current realm

36
Ticket and authenticator do not match

37
Clock skew is too great

38
Incorrect network address

39
Protocol version mismatch

40
Invalid message type

41
Message stream has been modified

42
Message is out of order

44
Key version is not available

45
Service key is not available

46
Mutual authentication failed

47
Incorrect message direction

48
Alternative authentication method required

49
Incorrect message sequence number

50
Inappropriate checksum type

60
Generic error detected

61
Field is too long

62
Client certificate is not acceptable

63
KDC certificate is not trusted or does not meet requirements

64
Certificate signature not valid

65
Client Diffie-Hellman key parameters not accepted

70
Client certificate could not be verified

71
Client certificate chain validation error occurred

72
Client certificate chain contains a revoked certificate

73
Revocation status for the certificate chain could not be determined

75
Kerberos client name does not match name bound to the client certificate

76
Kerberos KDC name does not match name bound to the KDC certificate

77
Key purpose restricts certificate usage

78
Certificate signature digest algorithm is not supported

79
PKAuthenticator is missing the required paChecksum

80
The signedData digest algorithm is not supported

81
The Public Key encryption delivery method is not supported

Security server operator commands

The operator commands in this section are supported by Network Authentication Service for z/OS (SKRBKDC started task).

All principal names and passwords should consist only of characters from the POSIX portable character set, but should not include any variant characters, such as brace, bracket, or currency symbols. For more information on principal names, refer to "Adding principals" on page 47.

If for some reason you need to use blanks in a principal name as part of a command-line argument, enclose the whole name in quotation marks.

If you have a need to use principal names or passwords that contain characters not in the POSIX portable character set (in other words, for national language reasons), be sure that the LANG value for the SKRBKDC started task is set to a code page that translates the national language characters to those that Resource Access Control Facility (RACF) can use. This applies to z/OS clients as well.

F SKRBKDC,parameters

Causes a command to be executed by the security server. This command is the same as MODIFY SKRBKDC.

Format

```
F SKRBKDC,parameters
```

Options

DISABLE ADMIN
Disables the Kerberos administration service. No changes can be made to the Kerberos database while the administration service is disabled.

DISPLAY ADMIN
Displays the current status of the Kerberos administration service. This option may be abbreviated D ADMIN.

DISPLAY CREDS,*owner*,*date*
Displays all credentials data space allocations for a user that were created before the specified date. All data space allocations for a user are displayed if the date is omitted. All data space allocations are displayed if no owner is specified. A date can be specified without specifying an owner by using two successive commas. A maximum of 252 allocations can be displayed. The date is specified as *yyyy.ddd*.

This command can be abbreviated as D CREDS,*owner,date*.

DISPLAY CRYPTO
Displays the available encryption types, whether hardware cryptographic support is available, and whether the encryption type can be used for application data. This option may be abbreviated D CRYPTO.

DISPLAY LEVEL
Displays the current service level of the Kerberos security server. This option may be abbreviated D LEVEL.

DISPLAY NETWORK
Displays the status of the network interfaces. The SKDC_NETWORK_POLL environment variable determines how often the Kerberos security server updates the network interface status. This option may be abbreviated D NETWORK.

DISPLAY PROP
Displays the status of database propagation. The current update sequence number is displayed for each Kerberos security server in the realm that participates in database propagation. This information is available only on the primary security server for the realm. This option may be abbreviated D PROP.

DISPLAY XCF
Displays the status of all instances of the SKRBKDC started task in the sysplex.

This command can be abbreviated as D XCF.

DEBUG ON
Enables debug mode for the SKRBKDC started task using the current subcomponent debug values.

DEBUG OFF
Disables debug mode for the SKRBKDC started task.

DEBUG *subcomponent.level,subcomponent.level,...*
Sets the SKRBKDC started task debug level for one or more subcomponents. All subcomponents can be changed by specifying an asterisk for the subcomponent. The debug level must be an integer between 0 and 9. All debug messages are suppressed for a subcomponent when its debug level is 0. The initial debug settings are obtained from the SKRBKDC environment variable file (**/etc/skrb/home/kdc/envar**).

For example, `F SKRBKDC,DEBUG *.1,KRB_KDC.8`

This sets debug level to 1 for all subcomponents and then sets the debug level to 8 for the KRB_KDC subcomponent.

Note: IBM does not recommend debug tracing to be running indefinitely due to potential performance impacts.

ENABLE ADMIN
Enables the Kerberos administration service. The administration service can be enabled only if the Kerberos database supports the administration function.

PROP *secondary-name*
Initiates a full database propagation to the specified secondary Kerberos security server. This command can be issued only on the primary Kerberos security server. The secondary security server must be defined in the **/etc/skrb/home/kdc/kpropd.acl** configuration file.

Usage

For the option *subcomponent.level,subcomponent.level,...*, all subcomponents can be changed by specifying an asterisk for the subcomponent. The debug level must be an integer between 0 and 9. All debug messages are suppressed for a subcomponent when its debug level is 0. The initial debug settings are obtained from the SKRBKDC environment variable file,**/etc/skrb/home/kdc/envar**.

Examples

To enable debug mode:

```
F SKRBKDC,DEBUG ON
```

To disable debug mode:

```
F SKRBKDC,DEBUG OFF
```

To set the debug level for one or more subcomponents:

```
F SKRBKDC,DEBUG *.1,KRB_KDC.8
```

This sets the debug level to 1 for all subcomponents and then sets the debug level to 8 for the KRB_KDC subcomponent.

MODIFY SKRBKDC,parameters

This command is the same as F SKRBKDC.

Format

MODIFY SKRBKDC,*parameters*

Options

Same as the F SKRBKDC,*parameters* command.

Usage

Same as F SKRBKDC,*parameters*.

Examples

Same as F SKRBKDC, *parameters*.

CTRACE debugging utility

Component trace records can be captured using either in-storage wrap buffers or an external writer. The component trace records contain the same information as the debug messages. The component trace options parameter can be used to set or modify the initial subcomponent debug levels. The MODIFY SKRBKDC,DEBUG command can then be used to change the subcomponent levels once the trace has been started.

The following commands start the component trace (CTIKDC00 is a sample parmlib member):

```
TRACE CT,WTRSTART=SKRBWTR
TRACE CT,ON,COMP=SKRBKDC,PARM=CTIKDC00
```

The following commands stop the component trace:

```
TRACE CT,OFF,COMP=SKRBKDC
TRACE CT,WTRSTOP=SKRBWTR
```

IPCS can be used to format and display the component trace records that were written to the trace datasets or contained in a dump dataset. The trace entry type codes are the same as the subcomponent names.

P SKRBKDC

Causes an orderly shutdown of the security server. This command is the same as STOP SKRBKDC.

Format

```
P SKRBKDC
```

There are no parameters (options).

STOP SKRBKDC

Causes an orderly shutdown of the security server. This command is the same as P SKRBKDC.

Format

```
STOP SKRBKDC
```

There are no parameters (options).

Kerberos administration server

The Kerberos administration server is provided as part of the SKRBKDC started task. The administration capabilities are dependent upon the Kerberos database selected by the SKDC_DATABASE environment

variable. Communication between the administration client and the administration server uses a variant of Sun RPC with GSS-API authentication. The **kadmin** command is provided to perform Kerberos administration functions. In addition, the **kadm5_*** API is provided for use by application programs.

z/OS Network Authentication Service now provides an administration server for the NDBM database. Administration for the SAF database is performed using the native system commands. The **kadmin** command and the **kadm5_*** API can be used with any Kerberos administration server that is compatible with Version 2 of the MIT Kerberos administration protocol.

Administration privileges

Authorization controls are provided through the **/etc/skrb/home/kdc/kadm5.acl** file. This file controls the Kerberos administration privileges. Each line represents a single administration access definition, has a maximum length of 255 characters, and is assumed to be in the code page specified by the LANG environment variable. Comment lines start with a semi-colon and blank lines are ignored. Each line consists of 2 fields: the client principal name and the privileges granted. The order of the lines in the file is important because the search stops as soon as a match is found for the principal making an administration request.

The client principal name can contain the following wildcards:

- ? represents a single character
- * represents zero or more characters
- Paired [ ] represent any one of the characters between the brackets.

The following administration privileges are defined. Use lowercase letters to define the granted privileges (any privilege not listed is denied) and use uppercase letters to define the denied privileges (any privilege not listed is granted). Do not mix uppercase and lowercase letters in the same definition.

Letter to use (uppercase or lowercase)	Privilege (granted or denied)
a	ADD is granted
A	ADD is denied
c	CHANGEPW is granted
C	CHANGEPW is denied
d	DELETE is granted
D	DELETE is denied
g	GET is granted (this may also be specified as i)
G	GET is denied (this may also be specified as I)
l	LIST is granted
L	LIST is denied
m	MODIFY is granted
M	MODIFY is denied
s	SETKEY is granted
S	SETKEY is denied
*	All privileges are granted

Administration RPC functions

- CHPASS_PRINCIPAL - Change the password for a principal.

This function requires CHANGEPW authority or the principal entry must be the authenticated client entry. The new password is subject to the minimum password lifetime, minimum password classes, and minimum password length rules in effect for the Kerberos database. Depending upon the database implementation, existing keys are deleted when the password is changed.

- CHPASS_PRINCIPAL3 - Change the password for a principal.

  This function is the same as the CHPASS_PRINCIPAL function with the addition that the key types and salt types can be specified. Depending upon the database implementation, existing keys can either be retained or deleted when the password is changed.

- CHRAND_PRINCIPAL - Generate random keys for a principal.

  This function requires CHANGEPW authority or the principal entry must be the authenticated client entry. The password change is subject to the minimum password lifetime rule in effect for the Kerberos database. Depending upon the database implementation, existing keys are deleted when the random keys are generated.

- CHRAND_PRINCIPAL3 - Generate random keys for a principal.

  This function is the same as the CHRAND_PRINCIPAL function with the addition that the key types and salt types can be specified. Depending upon the database implementation, existing keys can either be retained or deleted when the random keys are generated.

- CREATE_POLICY - Create an administration policy.

  This function requires ADD authority. The maximum length of a policy name is 128 characters. The name must consist of displayable graphic characters as determined by the locale in effect for the administration server. The name may not contain the backslash character.

  The following mask flags are supported:

  - KADM5_POLICY - Policy name supplied (required)
  - KADM5_PW_MIN_LIFE - Minimum password lifetime supplied
  - KADM5_PW_MAX_LIFE - Maximum password lifetime supplied
  - KADM5_PW_MIN_LENGTH - Minimum password length supplied
  - KADM5_PW_MIN_CLASSES - Minimum number of password classes supplied
  - KADM5_PW_HISTORY_NUM - Number of password history entries supplied

- CREATE_PRINCIPAL - Create a principal.

  This function requires ADD authority. The maximum length of a principal name is 235 characters, including the realm name and separator. The name must consist of displayable graphic characters as determined by the locale in effect for the administration server. The name may not contain the backslash or commercial at-sign characters.

  The following principal attributes are supported:

  - KRB5_KDB_DISALLOW_POSTDATED - Disallow post-dated tickets
  - KRB5_KDB_DISALLOW_FORWARDABLE - Disallow forwardable tickets
  - KRB5_KDB_DISALLOW_TGT_BASED - Disallow TGT-based tickets
  - KRB5_KDB_DISALLOW_RENEWABLE - Disallow renewable tickets
  - KRB5_KDB_DISALLOW_PROXIABLE - Disallow proxiable tickets
  - KRB5_KDB_DISALLOW_DUP_SKEY - Disallow duplicate session keys
  - KRB5_KDB_DISALLOW_ALL_TIX - Disallow all tickets
  - KRB5_KDB_REQUIRES_PRE_AUTH - Requires preauthentication
  - KRB5_KDB_REQUIRES_HW_AUTH - Requires hardware authentication
  - KRB5_KDB_REQUIRES_PWCHANGE - Requires password change
  - KRB5_KDB_DISALLOW_SVR - Disallow service tickets
  - KRB5_KDB_PWCHANGE_SERVICE - This is a password change service

The following mask flags are supported:

- KADM5_ATTRIBUTES - Principal attributes supplied
- KADM5_KVNO - Initial key version number supplied
- KADM5_MAX_LIFE - Maximum ticket lifetime supplied
- KADM5_MAX_RLIFE - Maximum renewable ticket lifetime supplied
- KADM5_POLICY - Policy name supplied
- KADM5_PRINC_EXPIRE_TIME - Account expiration time supplied
- KADM5_PRINCIPAL - Principal name supplied (required)
- KADM5_PW_EXPIRATION - Password expiration time supplied
- KADM5_TL_DATA - Tagged data supplied (the tagged data type must be greater than 255)

- CREATE_PRINCIPAL3 - Create a principal.

  This function is the same as the CREATE_PRINCIPAL function with the addition that the key types and salt types can be specified.
- DELETE_POLICY - Delete an administration policy.

  This function requires DELETE authority. An error is returned if the policy is still refered to by Kerberos principals.
- DELETE_PRINCIPAL - Delete a principal.

  This function requires DELETE authority.
- GET_POLICY - Get an administration policy.

  This function requires GET authority.
- GET_POLS - List the administration policy names.

  This function requires LIST authority. An error is returned if there are more than 1000 matches for the search expression.
- GET_PRINCIPAL - Get a principal.

  This function requires GET authority.
- GET_PRINCS - List the principal names.

  This function requires LIST authority. An error is returned if there are more than 1000 matches for the search expression.
- GET_PRIVS - Get administration privileges for the authenticated client.

  This function can be issued by any client. The privileges are obtained by matching the authenticated client name to entries in the **/etc/skrb/home/kdc/kadm5.acl** control file.
- MODIFY_POLICY - Modify an administration policy.

  This function requires MODIFY authority.

  The following mask flags are supported:

  - KADM5_PW_MIN_LIFE - Minimum password lifetime supplied
  - KADM5_PW_MAX_LIFE - Maximum password lifetime supplied
  - KADM5_PW_MIN_LENGTH - Minimum password length supplied
  - KADM5_PW_MIN_CLASSES - Minimum number of password classes supplied
  - KADM5_PW_HISTORY_NUM - Number of password history entries supplied
- MODIFY_PRINCIPAL - Modify a principal.

  This function requires MODIFY authority. Only the maximum ticket lifetime and the maximum renewable ticket lifetime values can be modified for protected principals (the architected Kerberos principals for the realm).

The following principal attributes are supported:

- KRB5_KDB_DISALLOW_POSTDATED - Disallow post-dated tickets
- KRB5_KDB_DISALLOW_FORWARDABLE - Disallow forwardable tickets
- KRB5_KDB_DISALLOW_TGT_BASED - Disallow TGT-based tickets
- KRB5_KDB_DISALLOW_RENEWABLE - Disallow renewable tickets
- KRB5_KDB_DISALLOW_PROXIABLE - Disallow proxiable tickets
- KRB5_KDB_DISALLOW_DUP_SKEY - Disallow duplicate session keys
- KRB5_KDB_DISALLOW_ALL_TIX - Disallow all tickets
- KRB5_KDB_REQUIRES_PRE_AUTH - Requires preauthentication
- KRB5_KDB_REQUIRES_HW_AUTH - Requires hardware authentication
- KRB5_KDB_REQUIRES_PWCHANGE - Requires password change
- KRB5_KDB_DISALLOW_SVR - Disallow service tickets
- KRB5_KDB_PWCHANGE_SERVICE - This is a password change service

The following mask flags are supported:

- KADM5_ATTRIBUTES - Principal attributes supplied
- KADM5_FAIL_AUTH_COUNT - Failed authentication count supplied
- KADM5_KVNO - Key version number supplied
- KADM5_MAX_LIFE - Maximum ticket lifetime supplied
- KADM5_MAX_RLIFE - Maximum renewable ticket lifetime supplied
- KADM5_POLICY - Policy name supplied
- KADM5_POLICY_CLR - No policy is associated with the principal
- KADM5_PRINC_EXPIRE_TIME - Account expiration time supplied
- KADM5_PW_EXPIRATION - Password expiration time supplied
- KADM5_TL_DATA - Tagged data supplied (the tagged data type must be greater than 255)

- RENAME_PRINCIPAL - Rename a principal.

  This function requires ADD and DELETE authority.
- SETKEY_PRINCIPAL - Set the encryption keys for a principal.

  This function requires SETKEY authority. The password change is subject to the minimum password lifetime rule in effect for the Kerberos database. Depending upon the database implementation, existing keys are deleted when the new keys are set.
- SETKEY_PRINCIPAL3 - Set the encryption keys for a principal.

  This function is the same as the SETKEY_PRINCIPAL function with the addition that the salt types can be specified. Depending upon the database implementation, existing keys can either be retained or deleted when the new keys are set.

Kerberos database propagation

The Kerberos security server supports two types of security registries: SAF and NDBM. The SAF registry stores Kerberos information in the z/OS system security database and uses SAF services to interface with the external security manager. The external security manager is responsible for database propagation between systems in the same sysplex and between systems in different sysplexes. Kerberos database propagation is not used in this environment and does not need to be configured.

The NDBM registry uses the POSIX database support provided by Unix System Services. The database files are located in the **/var/skrb/krb5kdc** directory. Kerberos database propagation is used to synchronize these files between systems in the same sysplex and between systems in different sysplexes.

The file system containing the **/var/skrb/krb5kdc** directory must be large enough to contain two copies of the registry database files plus a complete database dump file.

The Kerberos security server supports two database propagation protocols: full replacement and individual updates. The full replacement protocol sends the entire Kerberos database to each secondary Kerberos security server. This is the only propagation protocol supported by MIT Kerberos. The propagation occurs at timed intervals specified by the SKDC_KPROP_INTERVAL environment variable. A propagation does not occur if there have been no changes to the database since the last database propagation.

The individual update protocol sends just the database updates to each secondary Kerberos security server. The propagation occurs as each change is made to the database. The primary security server keeps track of the update level of each secondary security server and holds pending updates for an unavailable secondary server until the server becomes available. The individual update protocol should be used if it is supported by the primary KDC and the secondary KDC, since it performs much better than the full replacement protocol for large databases.

The **/etc/skrb/home/kdc/kpropd.acl** configuration file contains an entry for each Kerberos security server in the realm, and it specifies the role assigned to each of the servers. Each line consists of three fields, blank lines are ignored,comment lines are indicated by a semi-colon in the first position, and the file is assumed to be in the code page specified by the LANG environment variable. The maximum line length is 255 characters. The first field specifies the host name and optional port, separated by a colon, of a Kerberos security server. Port 754 is used for database propagation if a port is not specified either in **kpropd.acl** or for the **krb5_prop** service. The host name is used as the name of the Kerberos security server in the propagation status database and is converted to lowercase. The second field specifies the role assigned to that security server. The third field specifies the encryption type for the session key in the service ticket. The encryption type field is optional and the default encryption type list obtained from the Kerberos configuration file is used if the field is omitted.

The roles are:

- Primary - This is the primary security server for the realm. It owns the Kerberos registry database and sends updates to the other security servers in the realm.
- Replace - This is a secondary security server that receives updates by replacing the entire registry database as part of each propagation cycle. The database propagation contains principal policy and password history information. The secondary KDC must be at the MIT Kerberos 1.2.2 level or later.
- Compat - This is a secondary security server that receives updates by replacing the entire registry database as part of each propagation cycle. The database propagation does not contain principal policy or password history information. This propagation format is supported by MIT Kerberos 1.2.1 and earlier. Note that the principal policy and password history information is lost if a database created using this propagation method is later used by the primary KDC for the realm.
- Update - This is a secondary security server that receives individual database updates.
- Manual - This is a secondary security server that receives updates manually when the security server PROP command is issued.

The **/etc/skrb/home/kdc/kpropd.acl** configuration file must exist on the primary system and on each secondary system if database propagation is going to be used. The KDC assumes it is the only KDC in the realm if this file is not found. The host names specified in the **kpropd.acl** file must be valid DNS names and each must be the primary name assigned to its host system. The KDC locates its own entry by using DNS services to translate the host name returned by the **gethostname()** function and then by searching for the translated name in the configuration file.

The **kpropd.acl** file on the primary system must contain an entry for each KDC in the realm, including the primary KDC. These entries define the secondary servers to receive database propagations from the primary KDC.

The **kpropd.acl** file on the secondary systems requires only the entry for the primary KDC, although other entries can be specified. A secondary KDC accepts updates only from servers identified in its **kpropd.acl** configuration file. The secondary KDC checks for an authorized server by using DNS services to translate

the remote IP address for a connection to a host name and then searching for the host name in the configuration file.

The KDC identified as the primary KDC provides Kerberos administration services by listening for requests on the administration and password change ports. The primary KDC does not listen for requests on the propagation port.

A KDC identified as a secondary KDC does not provide Kerberos administration services. It does not listen for requests on the administration or password change ports. A secondary KDC listens for propagation requests on the propagation port.

The Kerberos registry must contain a service principal for each system with a KDC, including the primary KDC. The principal name is **host/***primary-host-name* where *primary-host-name* is the primary DNS name for the system. For example, if a KDC is running on system **dcesec4.krb390.ibm.com**, the Kerberos registry must contain the principal **host/dcesec4.krb390.ibm.com**.

Each system running a secondary KDC must have a **/var/skrb/krb5kdc/kpropd.ktf** key table file. This key table contains the host key for that system and is created using the **kadmin ktadd** subcommand. For example, if a secondary KDC is running on system **dcesec7.krb390.ibm.com**, the following **kadmin** commands should be issued on the **dcesec7.krb390.ibm.com** system:

```
addprinc host/dcesec7.krb390.ibm.com
ktadd -k /var/skrb/krb5kdc/kpropd.ktf host/dcesec7.krb390.ibm.com
```

The following is a sample **kpropd.acl** for a Kerberos realm containing three Kerberos security servers. Systems **dcesec4.krb390.ibm.com** and **dcesec7.krb390.ibm.com** are z/OS systems in the same sysplex, while system **dcecpt.mitkrb.ibm.com** is an AIX® system running MIT Kerberos. The KDC on **dcesec4.krb390.ibm.com** is the primary KDC for the realm. This sample configuration file can be found in **/usr/lpp/skrb/examples/kpropd.acl**:

```
; Sample kpropd.acl configuration file
;
; Host                            Role        Encryption type
; ----                            ----        ---------------
  dcesec4.krb390.ibm.com:754      Primary
  dcesec7.krb390.ibm.com:754      Update
  dcecpt.mitkrb.ibm.com:754       Replace     des-cbc-crc
```

The **kpropd.acl** file may be changed while the Kerberos security server is running and the changes will be picked up at the next propagation interval. However, the role of a security server may not be changed from primary to secondary or from secondary to primary while the security server is running.

Note: It is recommended that when configuring primary and secondary Kerberos security servers on z/OS, that the primary Kerberos security server be configured on a z/OS system that is running the lowest release of z/OS. This will ensure that new function is not enabled on a primary security server that is not supported on a secondary system. (for example, a new encryption type used for the NDBM master key and history key that is not supported on the lower z/OS release).

Setting up a secondary KDC

A new secondary KDC is added to the realm by performing the following steps:

1. Use the **kdb5_ndbm stash** command to create the database master key stash file on the secondary system.
2. Create the **/var/skrb/krb5kdc/kpropd.ktf** key table on the secondary system. Use the **kadmin ktadd** subcommand to add an entry for the secondary system host principal. The **kadmin** command must either be issued on the secondary system or the **kpropd.ktf** file must be copied to the secondary system.
3. Create the **/etc/skrb/home/kdc/kpropd.acl** configuration file on the secondary system. The primary KDC system for the realm must be listed. The secondary KDC systems can also be included to make it easier to move the primary KDC in the future.

4. Start the **kpropd** command on the secondary system. The command waits to receive a database propagation from the primary KDC.
5. Edit the **/etc/skrb/home/kdc/kpropd.acl** configuration file on the primary system. Add the new secondary system and specify **Manual** for the role.
6. Use the security server PROP command to send the database to the waiting **kpropd** command.
7. Start the SKRBKDC started task on the new secondary system after the **kpropd** command has completed.
8. Edit the **/etc/skrb/home/kdc/kpropd.acl** configuration file on the primary system. Change the role of the secondary system to **Update**, **Replace**, or **Compat**. You should always specify **Update** if the secondary security server supports the update propagation protocol.

Note: It is recommended that when configuring primary and secondary Kerberos security servers on z/OS, that the primary Kerberos security server be configured on a z/OS system that is running the lowest release of z/OS. This will ensure that new function is not enabled on a primary security server that is not supported on a secondary system. (for example, a new encryption type used for the NDBM master key and history key that is not supported on the lower z/OS release)

Moving the primary KDC to another system

The roles of the primary KDC and a secondary KDC can be swapped by performing the following steps:

1. Use the **kadmin ktadd** command to create the **/var/skrb/krb5kdc/kpropd.ktf** key table on the primary system if it does not already exist. This key table is used by a secondary KDC when it receives a database propagation from the primary KDC. Since the primary KDC is going to become a secondary KDC, it now needs this key table.
2. Use the security server DISABLE ADMIN command to freeze the database on the primary system.
3. Use the security server PROP command to send the current database to the secondary KDC.
4. Shut down the old and the new primary security servers.
5. Update the **/etc/skrb/home/kdc/kpropd.acl** control files and change the role of the old primary KDC to **Compat**, **Replace** or **Update** and change the role of the new primary KDC to **Primary**.
6. Restart both security servers.

Interoperability with MIT Kerberos

A z/OS security server can be a primary KDC or a secondary KDC in the same realm with MIT Kerberos (and compatible) security servers. The MIT Kerberos server must be at release 1.2.2 or later, if the z/OS security server is the primary KDC and the **Replace** role is used. The MIT Kerberos server must be at release 1.0 or later if the z/OS security server is the primary KDC and the **Compat** role is used. The MIT Kerberos server must be at release 1.0 or later if the z/OS security server is a secondary KDC.

If the z/OS security server is the primary KDC for the realm, then each MIT Kerberos security server must be listed in the **kpropd.acl** configuration file with the **Replace** or **Compat** propagation protocol. The **/etc/inetd.conf** configuration file on the secondary system must be updated to start the **kpropd** command when a propagation request is received from the primary KDC.

If the z/OS security server is a secondary KDC for the realm with an MIT Kerberos security server as the primary KDC, then the **Replace** protocol is always used. There is no need to update **/etc/inetd.conf** on the z/OS system since the SKRBKDC started task listens for database propagation requests. The **kpropd** command is used on the z/OS system to receive the initial database propagation before starting the SKRBKDC started task for the first time.

Refer to the MIT documentation for more information on setting up MIT Kerberos.

Chapter 4. RACF and z/OS Integrated Security Services Network Authentication Service

This topic provides information on using RACF with z/OS Integrated Security Services Network Authentication Service.

z/OS Network Authentication Service uses RACF to store and administer information about principals and realms, using RACF user profiles and general resource profiles. The KERB segment of the user profile is used to store information about z/OS Network Authentication Service principals on your local system. The general resource class KERBLINK allows you to map principals to RACF user IDs on your system. The general resource class REALM defines the local z/OS Network Authentication Service realm and its trust relationships with foreign realms.

RACF also provides a callable service named `R_ticketserv` (IRRSPK00) for application servers that use z/OS Network Authentication Service services. See *z/OS Security Server RACF Security Administrator's Guide* for more information.

This topic describes how to use RACF to complete the following steps in the implementation of z/OS Network Authentication Service.

1. Customizing the local environment.
   a. Defining your local RRSF (RACF remote sharing facility) node.
   b. Defining your local realm.
   c. Defining local principals.
2. Defining your foreign environment.
   a. Defining foreign realms.
   b. Mapping RACF user IDs for foreign principals.

For implementation details, see the additional information with this material, or *z/OS Integrated Security Services Network Authentication Service Programming*.

Customizing your local environment

Before beginning to create RACF definitions to support your z/OS Network Authentication Service implementation, you must define your local system as the local RRSF node. If you have already implemented RRSF, you should review the details in *z/OS Security Server RACF Security Administrator's Guide* before beginning to create RACF definitions for principals and realms.

In this topic, we will discuss defining your local server as the local realm. Once this definition is complete, you can begin defining your users as local principals and ensuring that their keys are registered with your local server.

Defining your local RRSF node

You must define your local system as the local RRSF node using the TARGET command, even if you are not planning to exploit RRSF functions. For example:

```
@TARGET NODE(ENDMVSA) LOCAL
```

You must define your local system as the local RRSF node to allow keys to be generated for local principals who have their passwords changed through application updates. If the local RRSF node is not defined, RACF will not generate keys for local principals who change their own passwords. See “Generating keys for local principals” on page 66 for more information.

If RRSF is already implemented, you might have already defined your local system as a local RRSF node. However, be sure to review the information in *z/OS Security Server RACF Security Administrator's Guide*.

Defining your local realm

You must define your local realm to RACF before you define local principals. This is because the local realm name is used to generate keys for local principals. You define your local realm by creating a profile in the REALM class called KERBDFLT. Using the KERB option of the RDEFINE and RALTER commands, you can specify the following information about your local realm:

KERBNAME
Name of the local realm.

MINTKTLFE
Minimum ticket lifetime for the local realm.

DEFTKTLFE
Default ticket lifetime for the local realm.

MAXTKTLFE
Maximum ticket lifetime for the local realm.

CHECKADDRS
KDC to check addresses in tickets as part of ticket validation processing. This should be disabled (default) if your requests pass through routers or firewalls using Network Address Translation (NAT).

ENCRYPT
Specifies which keys the realm is allowed to use. The supported key types are DES, DES3, DESD, AES128, AES 256, AES128SHA2 and AES256SHA2.

PASSWORD
Value of the password for the local realm.

Notes:

1. This password is not a RACF user password. Therefore, it is not constrained by SETROPTS password rules that can be specified to control user passwords. In addition, the installation-defined new-password exit (ICHPWX01) is not invoked.
2. A password value must be supplied. A 1-128 character password can be specified.
3. Uppercase and lowercase letters are accepted and maintained in the case in which they are entered.
4. ICSF must be available to set or change the password as the encryption keys for a REALM class profiles are generated using ICSF callable services. In addition, the user issuing the command may need to be permitted to the CSFOWH resource of the CSFSERVE class.

See *z/OS Security Server RACF Command Language Reference* for detailed information about using the KERB option of the RDEFINE and RALTER commands to administer profiles in the REALM class.

Important: If your installation shares the RACF database with systems running different releases of z/OS, administer local Network Authentication Service realms from *only* the highest level z/OS system. If you alter local realms from a lower level z/OS system, the realms might lose the use of z/OS Network Authentication Service keys supported on higher levels of z/OS. In addition, if you list realm information using the RLIST command on a lower level z/OS system, you might receive inconsistent information.

Example of defining the local realm

The following example shows a local realm KRB2000.IBM.COM being defined with a minimum ticket lifetime of 30 seconds, a default ticket lifetime of 10 hours, a maximum ticket lifetime of 24 hours, and a password of New744275Pw. All of the ticket lifetimes are specified in seconds. The administrator then lists the new REALM profile.

```
RDEFINE REALM KERBDFLT KERB(KERBNAME(KRB2000.IBM.COM) MINTKTLFE(30)
   DEFTKTLFE(36000) MAXTKTLFE(86400) PASSWORD(New744275Pw))
```

```
RLIST REALM KERBDFLT KERB NORACF
 CLASS      NAME
 -----      ----
 REALM      KERBDFLT

 KERB INFORMATION
 ----------------
 KERBNAME=  KRB2000.IBM.COM
 MINTKTLFE= 0000000030
 MAXTKTLFE= 0000086400
 DEFTKTLFE= 0000036000
 KEY VERSION= 001
 KEY ENCRYPTION TYPE= DES DES3 DESD AES128 AES256 AES128SHA2 AES256SHA2
 CHECK ADDRESSES= NO
```

See *z/OS Security Server RACF Command Language Reference* for RLIST authorization requirements.

If Kerberos is in use at your installation and you wish to define a realm name, you must consider the following. In order to help you distinguish between RACF and Kerberos REALM names since RACF allows differentiation in the nomenclature, you can use the RDEFINE command to define the non-Kerberos profile names in the REALM class. For example, if you want to define the SAFDFLT profile in the REALM class (as opposed to the KERBDFLT profile) using the APPLDATA field to define the RACF realm name, issue:

```
RDEFINE REALM SAFDFLT APPLDATA('racf.winmvs2c')
```

As a result, the realm name `racf.winmvs2c` is selected to give a name to the set of user IDs and other user information held in the security manager database.

Defining local principals

You must define local principals as RACF users using the ADDUSER and ALTUSER commands with the KERB option. This creates a KERB segment in the user profile. Each local principal must have a RACF password or password phrase. Therefore, do not use the NOPASSWORD option when defining local principals.

You can specify the following information for your local principals:

KERBNAME
Local principal name.

Note: Uppercase and lowercase letters are accepted and maintained in the case in which they are entered.

MAXTKTLFE
Maximum ticket lifetime for the local principal.

ENCRYPT
Specifies which keys the local principal is allowed to use. The supported key types are DES, DES3, DESD, AES128, AES256, AES128SHA2, and AES256SHA2.

Important: Whenever you update a local principal's ENCRYPT options, the principal's password or password phrase may require a change to ensure that a key of each type is generated and stored in the principal's user profile.

Example defining local principals:

```
ALTUSER LEMIEUX KERB(KERBNAME('JacquesLemieux'))
```

See *z/OS Security Server RACF Command Language Reference* for detailed information about using the KERB option of the ADDUSER and ALTUSER commands to administer user profiles for local principals.

Important: If your installation shares the RACF database with systems running different releases of z/OS, administer local Network Authentication Service principals from *only* the highest level z/OS system. If you alter local principals from a lower level z/OS system, the users might lose the use of z/OS Network Authentication Service keys supported on higher levels of z/OS. In addition, if you list user principal

information using the LISTUSER command on a lower level z/OS system, you might receive inconsistent information.

Generating keys for local principals

Local principals must have keys registered with the local z/OS Network Authentication Service server in order to be recognized as local principals. The user's definition as a local principal is not complete until the keys are generated. A key of each encryption type is generated from the local principal's RACF user password or password phrase and stored in the principal's user profile at the time of the user's password or password phrase change. If you want keys generated, be sure to use a password or password phrase change facility that will not result in an expired password that the user must change at next logon. For example, you can change a principal's password using the NOEXPIRED option of the ALTUSER command.

The use of each key is based on the z/OS Network Authentication Service configuration. See "Configuration of encryption types and FIPS level" on page 17 for more information on how to customize environment variables related to keys.

System considerations for key generation

In order to successfully complete password changes for key generation, the RACF address space must be started and must be executing under the authority of a user ID that is associated with a user identifier (UID). In addition, each user or administrator who generates keys must also have an associated UID. As an alternative to defining individual UIDs for each user ID, you can customize your z/OS UNIX security environment to allow default OMVS segment processing. For more information, see *z/OS Security Server RACF Security Administrator's Guide*. In addition, ICSF must be available in order to generate encryption keys for local principals and the user issuing the command may need to be permitted to the CSFOWH resource of the CSFSERV class. In some cases, such as TSO Logon, the password change runs in the RACF address space, so the user associated with the RACF address space may need access to the CSFOWH resource.

Important: If your installation shares the RACF database with systems running different releases of z/OS, ensure that user password or password phrase changes (for purpose of key generation) are executed from *only* the highest level z/OS system. If user passwords or password phrases are changed from a lower level z/OS system, users might lose the use of z/OS Network Authentication Service keys supported on higher levels of z/OS.

Methods for generating keys

Security administrator method: You can change a user's password so that a key can be generated using the ALTUSER command with the NOEXPIRED option. For example:

```
ALTUSER LEMIEUX PASSWORD(new1pw) NOEXPIRED
```

Notes:

1. Do not use the NOPASSWORD option.
2. You must specify a password value so a key can be generated.
3. All characters of the password will be translated to uppercase.

Alternatively, you can add a KERB segment for a local principal and generate the required key using a single RACF command. For example:

```
ALTUSER LEMIEUX PASSWORD(NEW1PW) NOEXPIRED KERB(KERBNAME('JacquesLemieux'))
```

You can also change a user's password phrase so that Kerberos keys are generated using the ALTUSER command with the NOEXPIRED option. For example:

```
ALTUSER LEMIEUX PHRASE(newphrase) NOEXPIRED
```

Notes:

1. The NOPHRASE option must not be used.

2. A password phrase value must be specified so a key can be generated.

For information about password phrase syntax rules, see *z/OS Security Server RACF Command Language Reference*.

You can also add a KERBNAME for a local principal and specify a password phrase, generating the required keys using a single RACF command. For example:

```
ALTUSER LEMIEUX PHRASE(newphrase) NOEXPIRED KERB(KERBNAME('JacquesLemieux'))
```

Note: Since a user can have both a password and a password phrase it is recommended that administrators avoid changing both PASSWORD and PHRASE on the same RACF command. Any uncertainty about which value was last used for Kerberos key generation can be handled by issuing a LISTUSER command.

User method: Users can change their own passwords, completing their own definitions as local principals, by using any standard RACF password-change facility, such as one of the following:

- TSO PASSWORD command (without the ID option)
- TSO logon
- CICS® signon

Users who have been assigned a password phrase can also generate Kerberos keys and complete their own definitions as local principals, by using any standard facility which can be used to change their RACF password phrase, such as one of the following:

- TSO PASSWORD (or PHRASE) command
- TSO logon
- **kpasswd** Kerberos command

Notes:

1. Password change requests from applications that encrypt the password prior to calling RACF will not result in usable keys.
2. Some applications may not be able to support password phrases. If using one of these applications results in a password change, that new password becomes the principal's Kerberos password and new Kerberos keys will be generated. For more information on password phrase syntax rules, see *z/OS Security Server RACF Command Language Reference*.

A local principal's key will be revoked whenever the user's RACF user ID is revoked or the RACF password or password phrase is considered expired. If the user's key is revoked, the server will reject ticket requests from this user.

Automatic local principal name mapping

For each local principal you define on your system using the KERB option of the ADDUSER and ALTUSER commands, RACF automatically creates a mapping profile in the KERBLINK class. When you issue the ALTUSER command with the NOKERB option or issue a DELUSER for a user with a KERB segment, RACF automatically deletes the KERBLINK profile.

The KERBLINK profile maps the local principal name to the user's RACF user ID. The name of the KERBLINK profile for a local principal is the principal name specified as the KERBNAME value with the ADDUSER or ALTUSER command.

Considerations for local principal names

The name of the KERBLINK profile contains the local principal name that is being mapped. Local principal names can contain imbedded blanks.

Blanks are not permitted as a part of a RACF profile name. Therefore, when building the KERBLINK profile name, as a result of specifying KERBNAME with the ADDUSER or ALTUSER command, RACF command processing will replace each blank with the X'4A' character (which often resolves to the ¢ symbol). This

can be seen in the output from the RLIST KERBLINK * command, and in the output from the RACF data base unload utility (IRRDBU00). RACF command processing also prevents the X'4A' character (¢) from being specified as part of the actual local principal name.

For information about the set of characters supported for local principal names, see *z/OS Security Server RACF Command Language Reference*.

Customizing your foreign environment

Your local z/OS Network Authentication Service server can trust authentications completed by other servers, and can be trusted by other servers, by participating in trust relationships. See the z/OS z/OS Network Authentication Service publications for details about trust relationships.

To participate in trust relationships, you must define each server as a foreign realm. Then, you can allow users who are authenticated in foreign realms (foreign principals) to access protected resources on your local system by mapping one or more RACF user IDs to foreign principal names. You do not need provide foreign principals ability to logon to your local system. You can simply provide mapping to one or more local user IDs so they can gain access privileges for local resources that are under the control of an application server, such as Db2®.RACF

Defining foreign realms

You define foreign realms by creating profiles in the REALM class. The profile name of the REALM class profile contains the fully qualified name of both servers in the relationship. The profile name uses the following formats:

```
/.../realm_1/KRBTGT/realm_2
```

Using the KERB option of the RDEFINE and RALTER commands, you define a REALM profile and specify the following information:

PASSWORD
Value of the password for this trust relationship with a foreign realm.

Notes:

1. This password is not the same as a RACF user password. Therefore, it is not constrained by the SETROPTS password rules that can be specified to control user passwords.
2. A value must be supplied to establish a trust relationship with this foreign realm. A 1-128 character password can be specified.
3. ICSF must be available to set or change the password as the encryption keys for a REALM class profile are generated using ICSF services. In addition, the user issuing the command may need to be permitted to the CSFOWH resource of the CSFSERV class.

See *z/OS Security Server RACF Command Language Reference* for detailed information about using the KERB option of the RDEFINE and RALTER commands to administer profiles in the REALM class.

For example:

```
RDEFINE REALM /.../KERBZOS.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM
    KERB(PASSWORD(1276458))
```

Mapping foreign principal names

You map foreign principals names to RACF user IDs on your local system by defining general resource profiles in the KERBLINK class. You can map each principal in a foreign realm to its own user ID on your local system, or you can map all principals in a foreign realm to the same user ID on your system.

RACF user IDs that map to foreign principals do not need KERB segments. These user IDs are intended to be used only to provide local RACF identities to associate with access privileges for local resources that are under the control of an application server, such as Db2.

Each mapping profile in the KERBLINK class is defined and modified using the RDEFINE and RALTER commands. The name of the KERBLINK profile for a foreign principal contains the principal name, fully qualified with the name of the foreign realm. The profile name uses the following format:

```
/.../foreign_realm/[foreign-principal_name]
```

If you wish to map a unique RACF user ID to each foreign principal, you must specify the foreign realm name and the foreign principal name. Note: generic characters (*, &, %) that are specified in a KERBLINK profile name are treated as non-generic characters since generic characters are disallowed for the KERBLINK class. If you wish to map the same RACF user ID to every foreign principal in the foreign realm, you need only specify the foreign realm name. In each case, you specify the local user ID using the APPLDATA option of the RDEFINE or RALTER command.

Note: If mapping a foreign principal that contains spaces, then the space characters will need to be replaced with the character mapped to the 0x4A hexadecimal value in the RDEFINE and RALTER commands. (In IBM code page 037 and 1047, 0x4A is the '¢' (cent sign), but in IBM code page 500 0x4A is the '[' (left bracket)).

Example of mapping foreign principal names

In the following example, the users SYKORA and Nedved will have their foreign principal names mapped with individual user IDs on the local z/OS system. All other foreign principals presenting tickets from the KERBZOS.ENDICOTT.IBM.COM server will be mapped to the ENDKERB user ID on the local z/OS system.

```
RDEFINE KERBLINK /.../KERBZOS.ENDICOTT.IBM.COM/SYKORA APPLDATA('PETRS')
RDEFINE KERBLINK /.../KERBZOS.ENDICOTT.IBM.COM/Nedved APPLDATA('PAVELN')
RDEFINE KERBLINK /.../KERBZOS.ENDICOTT.IBM.COM/       APPLDATA('ENDKERB')
```

Note: The characters of the profile name are *not* translated to uppercase so be sure to enter the realm portion of the profile name in *uppercase* and the foreign principal name in the appropriate case.

See *z/OS Security Server RACF Command Language Reference* for detailed information about using the RDEFINE and RALTER commands to administer mapping profiles for foreign principals in the KERBLINK class.

Part 2. Reference

This Reference section contains:

- Commands for Network Authentication Service for z/OS
- Status Codes for Network AuthenticationService for z/OS
- Messages for Network Authentication Service for z/OS
- Component trace for Network Authentication Service for z/OS

Chapter 5. Commands

This chapter presents Network Authentication Service for z/OS commands in alphabetical order. It provides the format, options, usage, and examples for each command.

The commands are installed in **/usr/lpp/skrb/bin**. In order to use these commands, you must update the PATH environment variable to place **/usr/lpp/skrb/bin** before **/bin** in the search order or else you must use the fully-qualified command name.

If you need to pass a parameter that includes a space or other special character in any of the commands then you will need to quote the argument or escape the special characters. For example:

```
kadmin -p fred -w "password with spaces"
```

or

```
kadmin -p fred -w password\ with\ spaces
```

kadmin

Administers the Kerberos database.

Format

```
kadmin [-r realm] [-p principal] [-k keytab] [-w password] [-A] [-e]
```

Options

-r *realm*
Specifies the Kerberos administration realm. If this option is not specified, the realm is obtained from the principal name. This option is meaningful only if the administration server supports multiple realms.

-p *principal*
Specifies the administrator principal. If this option is not specified, the string **/admin** is appended to the principal name obtained from the default credentials cache. If there is no credentials cache, the string **/admin** is appended to the name obtained from the USER environment variable, or, if the USER environment variable is not defined, it is appended to the name obtained from the **getpwuid()** function. The local realm is used if an explicit realm is not part of the principal name.

-k *keytab*
Specifies the key table containing the password for the administrator principal. The user is prompted to enter the password if neither the **-k** nor the **-w** option is specified. When using **-k**, the principal name is *host/host-name* unless the **-p** option is specified. The *host-name* is the primary host name for the local system.

-w *password*
Specifies the password for the administrator principal. The user is prompted to enter the password if neither the **-k** nor the **-w** option is specified.

-A
Specifies that the initial ticket used by the **kadmin** command does not contain a list of client addresses. If this option is not specified, the ticket contains the local host address list. When an initial ticket contains an address list, it can be used only from one of the addresses in the address list.

-e
Echoes each command line to **stdout**. This is useful when **stdout** is redirected to a file.

Usage

The **kadmin** command is used to manage entries in the Kerberos database. You are prompted to enter one or more subcommands. Each subcommand has a maximum length of 1023 characters. To enter a subcommand using multiple input lines, end each line to be continued with a backslash (\) character.

The **kadmin** command can be used with any Kerberos administration server supporting Version 2 of the Kerberos administration protocol. The z/OS Kerberos security server provides Kerberos administration server support for the NDBM database but not the SAF database (the normal system security commands are used to administer the SAF database).

The kadmin subcommands will now operate in FIPS compliance mode, when the KDC is running in FIPS mode (SKDC_FIPSLEVEL envar is set). Input parameters used for adding or modifying principal information will be evaluated for FIPS compliance when FIPS is enabled. The policies and attributes being associated with the principals will be evaluated for FIPS compliance when the Kerberos environment is running in FIPS mode.

Subcommand options start with a minus (-) character and principal attributes start with a plus (+) character or a minus (-) character. This means that principal and policy names must not start with these characters. In addition, since the backslash (\) character is used to indicate continuation and the single quote (') and double quote (") characters are used as delimiters, a name or password must not contain any of these characters. The **kadmin** command imposes no other restrictions on the characters used in names or passwords, although it is recommended that you do not use any of the EBCDIC variant characters. The Kerberos administration server may impose additional restrictions.

The following encryption types are supported by the **kadmin** command. An error is returned if an encryption type is specified that is not supported by the administration server.

- des-cbc-crc - DES encryption with 32-bit CRC checksum
- des-cbc-md4 - DES encryption with MD4 checksum
- des-cbc-md5 - DES encryption with MD5 checksum
- des-hmac-sha1 - DES encryption using key derivation and SHA1 checksum
- des3-cbc-sha1-kd - DES3 encryption using key derivation and SHA1 checksum
- aes128-cts-hmac-sha1-96 - AES128 encryption using key derivation and SHA1 checksum
- aes256-cts-hmac-sha1-96 - AES256 encryption using key derivation and SHA1 checksum
- aes128-cts-hmac-sha256-128 - AES128 encryption using key derivation and SHA2 checksum
- aes256-cts-hmac-sha384-192 - AES256 encryption using key derivation and SHA2 checksum

The following salt types are supported by the **kadmin command**. An error is returned if a salt type is specified that is not supported by the administration server.

- normal - Kerberos V5 salt using both the principal and realm names
- norealm - Kerberos V5 salt using just the principal name
- onlyrealm - Kerberos V5 salt using just the realm name
- afs3 - AFS® V3 salt
- v4 - Kerberos V4 salt

Time units

Dates are displayed as *day-of-week month day-of-month hour:minute:second timezone year* using the local timezone as specified by the TZ environment variable.

Durations are displayed as *days-hours:minutes:seconds*.

The **kadmin** command supports a number of date and duration formats, such as:

```
“15 minutes”
“7 days”
“1 month”
“2 hours”
```

```
“400000 seconds”
“next year”
“this Monday”
“next Monday”
yesterday
tomorrow
now
fortnight
“3/31/1992 10:00:07 PST”
“January 23, 2007 10:05pm”
“22:00 GMT”
2000-1-28
17-Jun.-2001
“28 Feb 2002"
```

The date specification must be enclosed in double quotes if it contains spaces. You cannot use a number without a unit (for example, “60 seconds” is correct but “60” is incorrect). If an explicit timezone is not given as part of the date specification, the local timezone is used. The date specification is not case sensitive; it may be entered using uppercase or lowercase characters. The year must be between 1970 and 2037. Two-digit years may be used with 0-37 representing 2000-2037 and 70-99 representing 1970-1999. The unit specification can be either singular or plural (for example, “month” and “months” are both allowed).

A date may be specified as an absolute or a relative value. If a relative value is given, the current date is added to form the date. An interval may also be specified as an absolute or a relative value. If an absolute value is given, the current date is subtracted to form the interval.

Here are the values that are acceptable for various ways of expressing time:

Units of time
year, month, fortnight, week, day, hour, minute, min, second, sec

Relative time
tomorrow, yesterday, today, now, last, this, next, ago

12-hour time delimiters
am, pm

Months
january, jan, february, feb, march, mar, april, apr, may, june, jun, july, jul, august, aug, september, sept, sep, october, oct, november, nov, december, dec

Days
sunday, sun, monday, mon, tuesday, tues, tue, wednesday, wednes, wed, thursday, thurs, thur, thu, friday, fri, saturday, sat

In addition, these time zones are acceptable:

*Table 9. Time zones recognized by the **kadmin** command*

Name	Description	Offset in Minutes	Daylight Savings Time Adjustment
GMT	Greenwich Mean	0	No
UT	Universal (Coordinated)	0	No
UTC	Universal (Coordinated)	0	No
WET	Western European	0	No
BST	British Summer	0	Yes
WAT	West Africa	60 West	No
AT	Azores	120 West	No
NFT	Newfoundland	210 West	No
NST	Newfoundland Standard	210 West	No

*Table 9. Time zones recognized by the **kadmin** command (continued)*

Name	Description	Offset in Minutes	Daylight Savings Time Adjustment
NDT	Newfoundland Daylight	210 West	Yes
AST	Atlantic Standard	240 West	No
ADT	Atlantic Daylight	240 West	Yes
EST	Eastern Standard	300 West	No
EDT	Eastern Daylight	300 West	Yes
CST	Central Standard	360 West	No
CDT	Central Daylight	360 West	Yes
MST	Mountain Standard	420 West	No
MDT	Mountain Daylight	420 West	Yes
PST	Pacific Standard	480 West	No
PDT	Pacific Daylight	480 West	Yes
YST	Yukon Standard	540 West	No
YDT	Yukon Daylight	540 West	Yes
HST	Hawaii Standard	600 West	No
HDT	Hawaii Daylight	600 West	Yes
CAT	Central Alaska	600 West	No
AHST	Alaska-Hawaii Standard	600 West	No
NT	Nome	660 West	No
IDLW	International Date Line West	720 West	No
CET	Central European	60 East	No
MET	Middle European	60 East	No
MEWT	Middle European Winter	60 East	No
MEST	Middle European Summer	60 East	Yes
SWT	Swedish Winter	60 East	No
SST	Swedish Summer	60 East	Yes
FWT	French Winter	60 East	No
FST	French Summer	60 East	Yes
EET	Eastern Europe	120 East	No
BT	Baghdad	180 East	No
IT	Iran	210 East	No
ZP4	Eastern Europe Zone 4	240 East	No
ZP5	Eastern Europe Zone 5	300 East	No
IST	Indian Standard	330 East	No
ZP6	Eastern Europe Zone 6	360 East	No

*Table 9. Time zones recognized by the **kadmin** command (continued)*

Name	Description	Offset in Minutes	Daylight Savings Time Adjustment
WAST	West Australian Standard	480 East	No
WADT	West Australian Daylight	420 East	Yes
JT	Java™	450 East	No
CCT	China Coast	480 East	No
JST	Japan Standard	540 East	No
KST	Korean Standard	540 East	No
CAST	Central Australian Standard	570 East	No
CADT	Central Australian Daylight	570 East	Yes
EAST	Eastern Australian Standard	600 East	No
EADT	Eastern Australian Daylight	600 East	Yes
GST	Guam Standard	600 East	No
KDT	Korean Daylight	600 East	No
NZT	New Zealand	720 East	No
NZST	New Zealand Standard	720 East	No
NZDT	New Zealand Daylight	720 East	Yes
IDLE	International Date Line East	720 East	No

Subcommands

The following subcommand descriptions assume the administration server is using the standard MIT Kerberos database for the registry. Other database implementations may not support all of the subcommand options and attributes.

The following subcommands are supported:

help [*subcommand*]
The **help** subcommand displays the command syntax for the specified subcommand. If no subcommand name is specified, the available subcommands are displayed.

get_privs
The **get_privs** (also known as **getprivs**) subcommand lists the administrative privileges for the authenticated client. Additional authorization checking may be performed for a specific administration function depending upon the function and the database implementation.

list_principals [*expression*]
The **list_principals** (also known as **listprincs**) subcommand lists all of the principals in the Kerberos database that match the specified search expression. If no search expression is provided, all principals are listed. You must have LIST authority.

The search expression can include the "*" and "?" wild cards where "*" represents zero or more characters and "?" represents a single character. For example, the expression ***/admin@*** returns all principal names that end with **/admin**, the expression **rwh*** returns all principal names that begin with **rwh**, and the expression **test_client?@*** returns principal names such as **test_client1**, **test_client2**, and so forth.

The search string can also contain paired "[" and "]" characters with one or more characters between the brackets. A match occurs if a name contains one of the characters between the brackets. For

example, the expression ***/[ad]*** returns all names containing **/a** and **/d**, while the expression **[ckr]*** returns all names beginning with **c**, **k**, or **r**.

get_principal *name*

The **get_principal** (also known as **getprinc**) subcommand displays information for a single principal entry including FIPS related principal key information.

Example:

```
getprinc test1

Principal: test1@ALPS4188.POK.IBM.COM
Number of keys: 1
…….
Key: Version 1, Type des-cbc-crc, Normal salt, Non-FIPS Compliant
……
```

The following principal attributes can be displayed by the **get_principal** subcommand. The attributes that are supported by the administration server are dependent upon the Kerberos database implementation.

DISALLOW_DUP_SKEY
Specifies that a service ticket cannot be encrypted using the session key of an existing ticket.

DISALLOW_FORWARDABLE
Specifies that forwardable tickets are not allowed.

DISALLOW_POSTDATED
Specifies that postdated tickets are not allowed.

DISALLOW_PROXIABLE
Specifies that proxiable tickets are not allowed.

DISALLOW_RENEWABLE
Specifies that renewable tickets are not allowed.

DISALLOW_SVR
Specifies that service tickets cannot be obtained for this principal.

DISALLOW_TGT_BASED
Specifies that service tickets cannot be obtained using a ticket-granting ticket.

DISALLOW_ALL_TIX
Specifies that tickets cannot be obtained for this principal.

REQUIRES_PWCHANGE
Specifies that the password must be changed.

PWCHANGE_SERVICE
Specifies that this is a password-changing service. The KDC grants an initial ticket to a password-changing service even if the current password is expired.

REQUIRES_HW_AUTH
Specifies that hardware authentication must be used when requesting a ticket. When requesting an initial ticket, hardware authentication must be used, and when requesting a service ticket, the ticket-granting ticket must indicate hardware authentication.

Note: z/OS does not support hardware authentication for an initial ticket but it will grant a service ticket when hardware authentication is requested if the ticket-granting ticket is from a foreign realm where hardware authentication is supported and used.

REQUIRES_PRE_AUTH
Specifies that preauthentication must be used when requesting a ticket. When requesting an initial ticket, preauthentication data must be provided, and when requesting a service ticket, the ticket-granting ticket must indicate preauthentication.

SUPPORT_DESMD5
Specifies that ENCTYPE_DES_CBC_MD5 keys are supported for this principal.

add_principal [*options*] [*attributes*] *name*

The **add_principal** (also known as **addprinc**) subcommand adds a new principal entry to the Kerberos database. The options and attributes may be specified before or after the principal name and may be entered in any order. You must have ADD authority.

The following options are supported for the **add_principal** subcommand:

-clearpolicy
Specifies that no policy is to be associated with the principal entry. The default policy is used if neither **-policy** nor **-clearpolicy** is specified and a policy named **default** exists. This option is mutually exclusive with the **-policy** option.

-e *key types*
Specifies the key types to be generated. Entries in the list are separated by commas. Each entry consists of an encryption type and a salt type, separated by a colon. The salt type can be omitted and defaults to **normal**. Similar encryption types are ignored when processing the list. For example, encryption types **des-cbc-crc** and **des-cbc-md5** use the same DES key, so only one of these encryption types needs to be specified to cause a DES key to be generated.

The keytypes are evaluated by the add_principal subcommand processing for FIPS compliance when FIPS is enabled (setting of fipslevel keyword in the Kerberos configuration file, /etc/skrb/krb5.conf by default).

When FIPS is enabled and the -e option is omitted, the key types which are FIPS compliant are obtained from the list of encryption types defined for the default_tkt_enctypes. If default_tkt_enctypes is not specified or if all the encryption types in the default_tkt_enctypes are not FIPS compliant then the product defined default encryption types **(aes256-cts-hmac-sha384-192, aes128-cts-hmac-sha256-128, aes256-cts-hmac-sha1-96, aes128-cts-hmac-sha1-96, des3-cbc-sha1)** will be used.

-expire *date*
Specifies the expiration date for the principal entry. If this option is not specified, the entry does not expire.

-kvno *version*
Specifies the key version number for the encryption keys generated by this command. If this option is not specified, the initial key version number is set to 1. A key version of 0 is not allowed.

-maxlife *interval*
Specifies the maximum ticket lifetime. If this option is not specified, the maximum ticket lifetime is obtained from the KDC policy.

-maxrenewlife *interval*
Specifies the maximum renewable ticket lifetime. If this option is not specified, the maximum renewable ticket lifetime is obtained from the KDC policy.

-policy *name*
Specifies the policy associated with the principal. The default policy is used if neither **-policy** nor **-clearpolicy** is specified and a policy named **default** exists. This option is mutually exclusive with the **-clearpolicy** option.

-pw *password*
Specifies the password for the principal entry. The user is prompted to enter the password in non-display mode if neither **-pw** nor **-randkey** is specified. This option is mutually exclusive with the **-randkey** option.

-pwexpire *date*
Specifies the expiration date for the password. If this option is not specified, the password lifetime from the effective policy is used to set the password expiration date.

-randkey
Specifies that a random key is to be generated for this principal. This option is mutually exclusive with the **-pw** option. If neither **-pw** nor **-randkey** is specified, the user is prompted to enter the password in non-display mode.

The following attributes are supported for the **add_principal** subcommand. The attributes that are supported by the administration server are dependent upon the Kerberos database implementation.

+allow_dup_skey
Specifies that a service ticket can be encrypted using the session key of an existing ticket. This is the default.

-allow_dup_skey
Specifies that a service ticket cannot be encrypted using the session key of an existing ticket.

+allow_forwardable
Specifies that forwardable tickets are allowed. This is the default.

-allow_forwardable
Specifies that forwardable tickets are not allowed.

+allow_postdated
Specifies that postdated tickets are allowed. This is the default.

-allow_postdated
Specifies that postdated tickets are not allowed.

+allow_proxiable
Specifies that proxiable tickets are allowed. This is the default.

-allow_proxiable
Specifies that proxiable tickets are not allowed.

+allow_renewable
Specifies that renewable tickets are allowed. This is the default.

-allow_renewable
Specifies that renewable tickets are not allowed.

+allow_svr
Specifies that service tickets can be obtained for this principal. This is the default.

-allow_svr
Specifies that service tickets cannot be obtained for this principal.

+allow_tgs_req
Specifies that service tickets can be obtained using a ticket-granting ticket. This is the default.

-allow_tgs_req
Specifies that service tickets cannot be obtained using a ticket-granting ticket.

+allow_tix
Specifies that tickets can be obtained for this principal. This is the default.

-allow_tix
Specifies that tickets cannot be obtained for this principal.

+needchange
Specifies that the password must be changed.

-needchange
Specifies that the password does not need to be changed. This is the default.

+password_changing_service
Specifies that this is a password changing service. The KDC grants an initial ticket to a password changing service even if the current password is expired.

-password_changing_service
Specifies that this is not a password changing service. This is the default.

+requires_hwauth
Specifies that hardware authentication must be used when requesting a ticket. Hardware authentication must be used when requesting an initial ticket, and the ticket-granting ticket must indicate hardware authentication when requesting a service ticket.

Note: z/OS does not support hardware authentication for an initial ticket but it will grant a service ticket when hardware authentication is requested if the ticket-granting ticket is from a foreign realm where hardware authentication is supported and used.

-requires_hwauth
Specifies that hardware authentication is not required. This is the default.

+requires_preauth
Specifies that preauthentication must be used when requesting a ticket. Preauthentication data must be provided when requesting an initial ticket, and the ticket-granting ticket must indicate preauthentication when requesting a service ticket.

Note that a z/OS KDC always requires preauthentication when requesting an initial ticket, even if this attribute is not set. This is done to improve the security of the Kerberos secret keys.

-requires_preauth
Specifies that preauthentication is not required. This is the default.

+support_desmd5
Specifies that ENCTYPE_DES_CBC_MD5 keys are supported for this principal.

When running in FIPS mode and the caller specifies this option, the command will fail since des-md5 is not a FIPS compatible encryption type.

-support_desmd5
Specifies that ENCTYPE_DES_CBC_MD5 keys are not supported for this principal. This is the default.

delete_principal *name*
The **delete_principal** (also known as **delprinc**) subcommand deletes a principal entry from the Kerberos database. You must have DELETE authority.

modify_principal [*options*] [*attributes*] *name*
The **modify_principal** (also known as **modprinc**) subcommand modifies an existing principal entry in the Kerberos database. The options and attributes may be specified before or after the principal name and may be entered in any order. You must have MODIFY authority.

The following options are supported for the **modify_principal** subcommand. The attributes that are supported by the administration server are dependent upon the Kerberos database implementation.

-clearpolicy
Specifies that no policy is to be associated with the principal entry. This option is mutually exclusive with the **-policy** option.

-expire *date*
Specifies the expiration date for the principal entry.

-kvno *version*
Specifies the key version number for the principal.

-maxlife *interval*
Specifies the maximum ticket lifetime.

-maxrenewlife *interval*
Specifies the maximum renewable ticket lifetime.

-policy *name*
Specifies the policy associated with the principal. This option is mutually exclusive with the **-clearpolicy** option.

-pwexpire *date*
Specifies the expiration date for the password.

The following attributes are supported for the **modify_principal** subcommand:

+allow_dup_skey
Specifies that a service ticket can be encrypted using the session key of an existing ticket. Resets the DISALLOW_DUP_SKEY attribute.

-allow_dup_skey
Specifies that a service ticket cannot be encrypted using the session key of an existing ticket. Sets the DISALLOW_DUP_SKEY attribute.

+allow_forwardable
Specifies that forwardable tickets are allowed. Resets the DISALLOW_FORWARDABLE attribute.

-allow_forwardable
Specifies that forwardable tickets are not allowed. Sets the DISALLOW_FORWARDABLE attribute.

+allow_postdated
Specifies that postdated tickets are allowed. Resets the DISALLOW_POSTDATED attribute.

-allow_postdated
Specifies that postdated tickets are not allowed. Sets the DISALLOW_POSTDATED attribute.

+allow_proxiable
Specifies that proxiable tickets are allowed. Resets the DISALLOW_PROXIABLE attribute.

-allow_proxiable
Specifies that proxiable tickets are not allowed. Sets the DISALLOW_PROXIABLE attribute.

+allow_renewable
Specifies that renewable tickets are allowed. Resets the DISALLOW_RENEWABLE attribute.

-allow_renewable
Specifies that renewable tickets are not allowed. Sets the DISALLOW_RENEWABLE attribute.

+allow_svr
Specifies that service tickets can be obtained for this principal. Resets the DISALLOW_SVR attribute.

-allow_svr
Specifies that service tickets cannot be obtained for this principal. Sets the DISALLOW_SVR attribute.

+allow_tgs_req
Specifies that service tickets can be obtained using a ticket-granting ticket. Resets the DISALLOW_TGT_BASED attribute.

-allow_tgs_req
Specifies that service tickets cannot be obtained using a ticket-granting ticket. Sets the DISALLOW_TGT_BASED attribute.

+allow_tix
Specifies that tickets can be obtained for this principal. Resets the DISALLOW_TIX attribute.

-allow_tix
Specifies that tickets cannot be obtained for this principal. Sets the DISALLOW_TIX attribute.

+needchange
Specifies that the password must be changed. Sets the REQUIRES_PWCHANGE attribute.

-needchange
Specifies that the password does not need to be changed. Resets the REQUIRES_PWCHANGE attribute.

+password_changing_service
Specifies that this is a password changing service. The KDC grants an initial ticket to a password changing service even if the current password is expired. Sets the PWCHANGE_SERVICE attribute.

-password_changing_service
Specifies that this is not a password changing service. Resets the PWCHANGE_SERVICE attribute.

+requires_hwauth
Specifies that hardware authentication must be used when requesting a ticket. Hardware authentication must be used when requesting an initial ticket, and the ticket-granting ticket must indicate hardware authentication when requesting a service ticket. Sets the REQUIRES_HW_AUTH attribute.

Note: z/OS does not support hardware authentication for an initial ticket but it will grant a service ticket when hardware authentication is requested if the ticket-granting ticket is from a foreign realm where hardware authentication is supported and used.

-requires_hwauth
Specifies that hardware authentication is not required. Resets the REQUIRES_HW_AUTH attribute.

+requires_preauth
Specifies that preauthentication must be used when requesting a ticket. Preauthentication data must be provided when requesting an initial ticket, and the ticket-granting ticket must indicate preauthentication when requesting a service ticket. Sets the REQUIRES_PRE_AUTH attribute.

-requires_preauth
Specifies that preauthentication is not required. Resets the REQUIRES_PRE_AUTH attribute.

Note that a z/OS KDC always requires preauthentication when requesting an initial ticket, even if this attribute is not set. This is done to improve the security of the Kerberos secret keys.

+support_desmd5
Specifies that ENCTYPE_DES_CBC_MD5 keys are supported for this principal. Sets the SUPPORT_DESMD5 attribute.

When running in FIPS mode and the caller specifies this option, the command will fail since des-md5 is not a FIPS compatible encryption type.

-support_desmd5
Specifies that ENCTYPE_DES_CBC_MD5 keys are not supported for this principal. Resets the SUPPORT_DESMD5 attribute.

change_password [-randkey | -pw *password*] [-keepold] [-e keytypes] *name*
The **change_password** (also known as **cpw**) subcommand changes the password for a principal. You must have CHANGEPW authority, or the principal entry must be your own entry.

A random key is generated if the **-randkey** option is specified. Otherwise, you are prompted to enter the new password unless the **-pw** option is specified.

Any existing encryption keys are discarded unless the **-keepold** option is specified. The number of retained keys is dependent upon the Kerberos database implementation.

The **-e** option is used to specify desired key types to be generated. Entries in the key types list are separated by commas. Each entry consists of an encryption type and a salt type, separated by a colon. The salt type can be omitted and defaults to **normal**. Similar encryption types are ignored when processing the list. For example, encryption types **des-cbc-crc** and **des-cbc-md5** use the same DES key, so only one of these encryption types needs to be specified to cause a DES key to be generated.

The keytypes are evaluated by the change_password subcommand processing for FIPS compliance when FIPS is enabled (setting of **fipslevel keyword in the Kerberos configuration file, /etc/skrb/krb5.conf by default**). When FIPS is enabled and the -e option is omitted, the key types that are FIPS compliant are obtained from the list of encryption types defined for the **default_tkt_enctypes**. If default_tkt_enctypes is not specified or if all the encryption types in the default_tkt_enctypes are not FIPS compliant, then the product defined default encryption types **(aes256-cts-hmac-sha384-192, aes128-cts-hmac-sha256-128, aes256-cts-hmac-sha1-96, aes128-cts-hmac-sha1-96, des3-cbc-sha1)** will be used.

When the password change request is received by the KDC, it will use the SKDC_FIPSLEVEL specified in the KDC envar file (/etc/skrb/home/kdc/envar by default) to determine whether the requested key types need to be verified to meet the given FIPS level. When FIPS is enabled and password history is being maintained for the given principal, duplicate password checking cannot be performed when the password history entries are non-FIPS compliant keys. When this occurs, a duplicate password will be accepted for a password history entry that utilizes a Non-FIPS compliant key. To identify principals that have non-FIPS compliant password history entries or principals that do not have any FIPS compliant current keys, run the **kdb5_ndbm utility fips_report** operation. If the duplicate password check could not be performed for one or more of the password history entries, an informational

message, **EUVF04175I,** is issued indicating the number of password history entries that could not be checked.

Note: If the database was created on z/OS 2.5 and is being used on z/OS 2.3, it is possible that the reuse of a previous password may not be detected.

rename_principal *oldname newname*
The **rename_principal** (also known as **renprinc**) subcommand changes the name of a principal entry in the Kerberos database. You must have both ADD and DELETE authority.

Since the principal name is often used as part of the password salt, you should change the password for the principal after the entry is renamed. Some implementations of the Kerberos administration server do not allow a principal to be renamed if the principal name is used in the password salt. In this case, you must delete the existing principal entry using the **delete_principal** subcommand and then add the new principal entry using the **add_principal** subcommand.

list_policies [*expression*]
The **list_policies** (also known as **listpols**) subcommand lists all of the policies in the Kerberos database that match the specified search expression. All policies are listed if no search expression is provided. You must have LIST authority.

The search expression can include the "*" and "?" wild cards where "*" represents zero or more characters and "?" represents a single character. For example, the expression ***_local** returns all policy names that end with **_local**, the expression **def*** returns all policy names that begin with **def**, and the expression **test_policy?** returns policy names such as **test_policy1**, **test_policy2**, and so forth.

The search string can also contain paired "[" and "]" characters with one or more characters between the brackets. A match occurs if a name contains one of the characters between the brackets. For example, the expression **[adh]*** returns all names beginning with **a**,**d**, or **h**.

get_policy *name*
The **get_policy** (also known as **getpol**) subcommand displays information for a single policy entry. You must have GET authority or the policy must be associated with your own principal entry.

add_policy [*options*] *name*
The **add_policy** (also known as **addpol**) subcommand adds a new policy to the Kerberos database. The options may be specified before or after the policy name and may be specified in any order. You must have ADD authority.

The following options are supported for the **add_policy** subcommand:

-maxlife *interval*
Specifies the maximum password lifetime. The password must be changed after this interval has elapsed.

-minlife *interval*
Specifies the minimum password lifetime. A new password cannot be changed until this interval has elapsed.

-minlength *number*
Specifies the minimum password length.

-minclasses *number*
Specifies the minimum number of character classes in the password.

-history *number*
Specifies the number of passwords in the password history. A new password cannot match any of the remembered passwords.

When a password change is requested for the principal, keys are generated with the new password. When the new password is not a duplicate, the strongest encryption type key generated for the new password is added to the principal password history entry.

When FIPS is enabled duplicate checking cannot be performed with Non-FIPS password history entries. A duplicate password will be accepted.

To identify principals that have Non-FIPS compliant password history keys or principals that do not have any FIPS compliant current keys, we are adding a new operation **fips_report** to the utility.

Note: If the database was created on z/OS 2.5 and is being used on z/OS 2.3, it is possible that the reuse of a previous password may not be detected.

modify_policy [*options*] *name*
The **modify_policy** (also known as **modpol**) subcommand modifies an existing policy in the Kerberos database. The options may be specified before or after the policy name and may be specified in any order. You must have MODIFY authority.

The following options are supported for the **modify_policy** subcommand:

-maxlife *interval*
Specifies the maximum password lifetime. The password must be changed after this interval has elapsed.

-minlife *interval*
Specifies the minimum password lifetime. A new password cannot be changed until this interval has elapsed.

-minlength *number*
Specifies the minimum password length.

-minclasses *number*
Specifies the minimum number of character classes in the password.

-history *number*
Specifies the number of passwords in the password history. A new password cannot match any of the remembered passwords.

delete_policy *name*
The **delete_policy** (also known as **delpol**) subcommand deletes a policy entry from the Kerberos database. You must have DELETE authority.

add_key [[-keytab | -k] *keytab_name*] [-keepold] [-e keytypes] *principal_name*
The **add_key** (also known as **ktadd**) subcommand generates a set of random encryption keys for the named principal and then adds the generated keys to the specified key table. You must have CHANGEPW authority or the principal entry must be your own entry.

The default key table is used if the **-keytab** option is not specified. A key table name prefix of “FILE:” is changed to “WRFILE:” because the **add_key** subcommand must update the key table.

Any existing encryption keys are discarded unless the **-keepold** option is specified. The number of retained keys is dependent upon the Kerberos database implementation.

All available key types are generated unless the **-e** option is used to specify desired key types to be generated. Each entry consists of an encryption type and a salt type, separated by a colon. The salt type can be omitted and defaults to **normal**. Similar encryption types are ignored when processing the list. For example, encryption types **des-cbc-crc** and **des-cbc-md5** use the same DES key, so only one of these encryption types needs to be specified to cause a DES key to be generated.

The keytypes are evaluated by the add_key subcommand processing for FIPS compliance when FIPS is enabled (setting of fipslevel keyword in the Kerberos configuration file, /etc/skrb/krb5.conf by default). When FIPS is enabled and the -e option is omitted, the key types that are FIPS compliant are obtained from the list of encryption types defined for the default_tkt_enctypes. If default_tkt_enctypes is not specified or if all the encryption types in the **default_tkt_enctypes** are not FIPS compliant, then the product defined default encryption types **(aes256-cts-hmac-sha384-192, aes128-cts-hmac-sha256-128, aes256-cts-hmac-sha1-96, aes128-cts-hmac-sha1-96, des3-cbc-sha1)** will be used.

kdb5_ndbm

This is the Kerberos NDBM database maintenance utility.

Format

```
kdb5_ndbm create [-k keytype] [-e keytypes]
kdb5_ndbm destroy
kdb5_ndbm dump [-k keytype] [-mkey_convert] [-hkey_convert] [-compat][-v] filename
kdb5_ndbm load [-k keytype] [-K keytype] [-mkey_convert] [-hkey_convert] [-v] filename
kdb5_ndbm stash [-k keytype]
kdb5_ndbm fips_report
```

Options

-e keytypes

Specifies the encryption types to be used when generating the initial principal keys. Keys are generated for all supported encryption types if this option is not specified. When FIPS is enabled using the fipslevel keyword in the krb5.conf Kerberos configuration file, the encryption types are validated for FIPS compliance. If this option is not specified when FIPS is enabled only FIPS compliant keys from the list of supportedencryption types are generated. Entries in the list are separated by commas. Similar encryption types are ignored when processing the list. For example, encryption types **des-cbc-crc** and **des-cbc-md5** use the same 56-bit DES key, so only one of these encryption types needs to be specified to cause a 56-bit DES key to be generated.

-k keytype

Specifies the encryption type for the database master key and will be validated for FIPS compliance when FIPS is enabled.

When the KDC is running in FIPS mode (SKDC_FIPSLEVEL is set in KDC envar file), the encryption type of the master key of the NDBM database will be validated for FIPS compliance. If the master key encryption type is not FIPS compliance, KDC will fail to start.

When specified on the **kdb5_ndbm create** command: The encryption type specified for the **-k** option is added to the encryption types specified for the **-e** option if it is not already in the list, because the **K/M** and **kadmin/history** architected principals must have a key available that is the same encryption type. The **aes256-cts-hmac-sha384-192** encryption type is used if -k option is not specified.

When specified on a **kdb5_ndbm stash** command: The keytype specifies the encryption type for the database master key. The **aes256-cts-hmac-sha384-192** encryption type is used if -k option is not specified.

When specified on a **kdb5_ndbm dump** command: The keytype specifies the encryption type for the database master key when the **-mkey_convert** option is also specified. Otherwise, the **-k keytype** option is ignored. The **aes256-cts-hmac-sha1-96** encryption type is used if -k option is not specified.

When specified on a **kdb5_ndbm load** command: The keytype specifies the master key encryption type in the dump file. If not specified, the dump key type will be determined from the dump file. If the given keytype does not match the master keytype in the dump file, the load request fails and the encryption type of the master key in dump file is returned back to the caller for retry.

-mkey_convert
Indicates that the master key is to be changed.

-hkey_convert
Indicates that the kadmin/history principal key is to be changed. This option is needed on a dump or load operation if the database master key type is changed and the kadmin/history principal does not have a key that is the same encryption type as the master key.

-compat
Creates the database dump using the version 4 format instead of the version 5 format.

-v
The principal and policy names should be displayed as they are processed.

-K <keytype>

Specifies the new database master key encryption type if the **-mkey_convert** option is specified on the kdb5_ndbm load command. Therefore, if a master key type change is desired, the **-mkey_convert** option must be specified on the **kdb5_ndbm load** command in conjunction with the **-K keytype** option. If this option is not specified, the encryption type of the new master key will be the same as the master key encryption type in the dump file.

Usage

The **kdb5_ndbm** command is used to maintain a Kerberos NDBM registry database. It is not used with a Kerberos SAF registry database. The **kdb5_ndbm** command must be run by a user with write access to the **/var/skrb/krb5kdc** directory and to all of the files in this directory. The **/var/skrb/krb5kdc** directory path is created if it does not already exist.

If FIPS mode is turned ON for KDC, ensure that the encryption type for the NDBM database master key is FIPS compliant.

The following functions are provided:

- **kdb5_ndbm create [-k keytype] [-e keytypes]**

  This command creates a new Kerberos NDBM database. An error is reported if an NDBM database already exists. The architected KDC principals are created in the new database. In addition, user principals **IBMUSER** and **IBMUSER/admin** are created with an initial password of IBMUSER. Finally, the master key stash file is created.

  The **kdb5_ndbm create** command is used to create a new database for the primary KDC for a realm. It is not used to create a database for a secondary KDC since a secondary KDC receives its database by propagation from the primary KDC.

  You are prompted to enter the master key for the new database. This key is used to encrypt the database entries and should not be an obvious password string. Do not forget this password since you need it when you create a secondary KDC or when you attempt to reload the database from a backup copy created by the **kdb5_ndbm dump** command.

- **kdb5_ndbm destroy**

  This command destroys an existing Kerberos NDBM database. The database files are removed along with the master key stash file.

- **kdb5_ndbm dump [-k keytype] [-mkey_convert] [-hkey_convert] [-compat] [-v] filename**

  This command creates a portable copy of the Kerberos NDBM database. The dump file is a printable text file created in the local code page as defined by the LANG environment variable. You should convert it to the code page of the target system when moving it to another system.

  The **kdb5_ndbm dump** command is used to create a backup copy of the Kerberos database. The database can be recreated from the backup copy using the **kdb5_ndbm load** command. The backup copy can also be used to create a copy of the Kerberos database on another system. The NDBM database files themselves cannot be moved to another system since the internal database formats are not portable.

  The database dump includes principal and policy information. The dump key is the same as the database master key unless the **-mkey_convert** option is specified. When **-mkey_convert** is specified, a new key is generated from the supplied dump password and will be the new master key for the dumped database. If the -k option is not specified, the generated master key will be an **aes256-cts-hmac-sha1-96** key. If the new master key is of a different encryption type than the previous master key, the **-hkey_convert** option will need to be specified if the kadmin/history principal does not have an encryption key that is the same type as the new master key. In this situation, if the **-hkey_convert** option is not specified on the **kdb5_ndbm dump** command, the dump operation will succeed, however, a subsequent **kdb5_ndbm load** operation will fail if the **-hkey_convert** option and possibly the -k option are not specified (if the master key type is not **aes256-cts-hmac-sha1-96**). It is therefore recommended that if a master key conversion is being performed on the dump operation and the encryption type of the master key is being changed, that the **-hkey_convert** option is also specified.

The dump file created by the z/OS Network Authentication Service **kdb5_ndbm** command uses the version 5 dump format and is compatible with the dump file created by the MIT Kerberos 1.2.2 **kdb5_util** command dump r13 format. The version 5 dump format includes principal policy and password history information. Earlier releases of the MIT Kerberos **kdb5_util** command do not support this dump format. You can specify the **-compat** option to create a dump in the version 4 format, which can be processed by earlier releases of MIT Kerberos. A version 4 dump does not include principal policy or password history information. This information is lost if a database created from a version 4 dump is used by the primary KDC for the realm.

- **kdb5_ndbm load [-k keytype] [-K keytype] [-mkey_convert] [-hkey_convert] [-v] filename**

  This command creates a Kerberos NDBM database using the portable copy created by the **kdb5_ndbm dump** command. An error is returned if an NDBM database already exists.

  You are prompted for the dump key. The master key for the database is the same as the dump key unless the **-mkey_convert** option is specified. If the -K option is not specified in conjunction with the **-mkey_convert** option, the generated master key will be the same as the master key type in the dump file, otherwise the master key will be the key type specified by the -K option. If the new master key is of a different type than the previous master key, the **-hkey_convert** option will need to be specified if the kadmin/history principal does not have an encryption key that is the same type as the new master key type. The **-hkey_convert** option is the only way to change the kadmin/history key and will replace the current kadmin/history principal keys with a single encryption key of the same type as the master key.

  The z/OS Network Authentication Service **kdb5_ndbm** command processes dump files created by MIT Kerberos 1.2 (version 4 dump format) and MIT Kerberos 1.2.2 (version 5 dump format r13). Per-principal policy information is lost if the version 4 dump format is used, because that dump format does not contain this information.

- **kdb5_ndbm stash [-k keytype]**

  This command creates the database master key stash file. An error is returned if the stash file already exists since the master key cannot be changed after the database is created. Use the **kdb5_ndbm destroy** command to remove an existing database before attempting to create a new database.

  The **kdb5_ndbm stash** command is used to create the master key stash file for a secondary KDC. An error occurs during database propagation if the stash file is created with the wrong database master key password or the wrong database master key type.

- **kdb5_ndbm fips_report**

  The kdb5_ndbm **fips_report** command is used to generate a report to lists the FIPS compliance status of principals' password history keys and current keys. Prior to enabling FIPS for an existing Network Authentication Service NDBM database, run the **kdb5_ndbm fips_report** utility to check for readiness to enable FIPS. The following describes actions that must be taken for each scenario:

  1. If the master key is not FIPS compliant, follow the kdb5_ndbm documentation for the dump and load operation to change the master key encryption type using -mkey_convert option. If the **kadmin/history** principal does not have FIPS compliant keys, update the history key when changing the master key encryption type using -hkey_convert option.
  2. For principals in the existing NDBM database that only have **DES and/or DESD keys**, enable FIPS compliant encryption type as documented in NAS Administration and change the password for the indicated principals. Re-run the utility to verify the issue is resolved.
  3. For principals that maintain password that have only DES history keys, be aware that password history checks during a **change_password** will be unable to generate a DES key from the new password when FIPS is enabled to compare with the history keys to detect a password re-use attempt. If this is a required security capability in your environment, configure the environment to use only FIPS compatible keys, but do not enable FIPS until there are no more DES history keys reported from the kdb5_ndbm **fips_report** utility.

  Sample Output:

  ```
  kdb5_ndbm fips_report

  Principal Name: test1@ALPS4188.POK.IBM.COM
  ```

```
EUVF04178E Principal contains only Non-FIPS compliant encryption keys.
EUVF04182W Principal history contains both FIPS and Non-FIPS compliant encryption keys.

Principal Name: test2@ALPS4188.POK.IBM.COM
EUVF04179W Principal contains both FIPS and Non-FIPS compliant encryption keys.
EUVF04181E Principal history contains only Non-FIPS compliant encryption keys.

Principal Name: test3@ALPS4188.POK.IBM.COM
EUVF04180I All principal keys are FIPS compliant.
EUVF04183I All principal history keys are FIPS compliant.

Messages will be generated for each principal based on the principal key encryption type and
principal history
encryption type. The message suffix E – Error, W – Warning, I – Informational will indicate
what further actions
that needs to be taken.
```

Example:

Principal test1@ALPS4188.POK.IBM.COM
EUVF04178E indicates that all principal keys are Non-FIPS compliant. Prior to enabling FIPS mode, enable
FIPS compliant encryption type as documented in NAS Administration and change the password for the
indicated principals. Re-run the utility to verify the issue is resolved.
EUVF04182W indicates that the principal history entry contains both FIPS and Non-FIPS compliant keys.
Prior to enabling FIPS mode, ensure there are no more DES history keys reported from the kdb5_ndbm
fips_report utility by changing the password for the indicated principals using the change_password kadmin
subcommand.

Principal **test2@ALPS4188.POK.IBM.COM**
EUVF04179W indicates that the principal keys are a mixture of FIPS and Non-FIPS compliant keys. Prior to enabling
FIPS mode, you may enable FIPS compliant encryption type as documented in NAS Administration and change the
password for the indicated principals. Re-run the utility to verify the issue is resolved. This is recommended action but
not a mandatory action.

EUVF04181E indicates that the principal history entry contains only Non-FIPS compliant keys. The password
history checks during change_password will be unable to generate a DES key from the new password when
FIPS is enabled to compare with the history keys to detect a password re-use attempt. Prior to enabling FIPS
mode, ensure there are no more DES history keys reported from the kdb5_ndbm **fips_report** utility by changing
the password for the indicated principals using the change-password kadmin subcommand.

kdestroy

Destroys a Kerberos credentials cache.

Format

```
kdestroy [-c cache_name] [-e time_delta]
```

Options

-c *cache_name*
Specifies the name of the credentials cache to destroy. The default credentials cache is destroyed if no command options are specified. This option and the **-e** option are mutually exclusive.

-e *time_delta*
Specifies that all credentials cache files containing expired tickets are deleted if the tickets have been expired at least as long as the *time_delta* value.

Usage

The kdestroy command deletes a Kerberos credentials cache file.

The **-e** option causes the **kdestroy** command to check all of the credentials cache files in the default cache directory (**/var/skrb/creds**). Any file that contains only expired tickets that have expired for the time delta are deleted. The time delta is expressed as *n***w***n***d**n**h***n***m***n***s** where *n* represents a number, **w** indicates weeks, **d** indicates days, **h** indicates hours, **m** indicates minutes, and **s** indicates seconds. The components must be specified in this order but any component may be omitted (for example, 4h5m represents four hours and 5 minutes and 1w2h represents 1 week and 2 hours). If only a number is specified, the default is hours.

To delete a credentials cache, the user must be the owner of the file or must be a root (uid 0) user.

Examples

To delete the default credentials cache for the user:

```
kdestroy
```

To delete all credentials caches with expired tickets older than 1 day:

```
kdestroy -e 1d
```

keytab

Manages a key table.

Format

```
keytab add principal [-r][-p password] [-v version] [-k keytab]
keytab check [principal] [-k keytab]
keytab delete principal [-v version] [-k keytab]
keytab list [principal [-v version]] [-k keytab]
keytab merge in_keytab [principal [-v version]] [-r] [-k keytab]
```

Options

-k *keytab*
Specifies the key table name. The default key table is used if this option is not specified.

-p *password*
Specifies the password. The user is prompted to enter the password if this option is not specified when adding an entry to the key table.

Note that the password provided for the keytab command must exactly match the password that was used to generate the key in the KDC. For example, this password is case sensitive. If your KDC uses a RACF database with mixed-case passwords disabled then this password needs to be entered in uppercase. If you are embedding spaces then they need to be escaped or the whole password needs to be quoted according to your shell rules.

-r
When specified on the keytab add, or merge options, entries whose principal name and version number are identical will be deleted from the target keytab file before the new entries for the given principal and version number are added.

-v *version*
Specifies the key version number. When adding a key, the next version number is assigned if this option is not specified. When deleting, listing or merging keytabs, all keys for the principal are deleted if this option is not specified. When the version number is specified, the principal name must also be specified.

in_keytab is the path of the keytab to be merged to the keytab file. It must be specified during a merge operation, and the file must exist.

The principal name is optional when merging, listing or checking a keytab. When the principal name is specified, only entries for that principal are operated on.

Usage

The **keytab** command is used to add or delete a key from a key table, display the entries in a key table or to merge the entries of two key tables. It can also be used to check keytab entries. Some operations may not be easily reversible without knowledge of matching passwords in the KDC, so you might want to save a copy of your keytab file, prior to performing the keytab command.

The **keytab** file contains multiple entries for the same principal and version number. These entries match the entries created in the KDC for different encryption types, provided the same case sensitive password was used to generate the keys during the creation of the keytab entries as was used when used adding or updating the principal's password in the KDC. Different version numbers may be used in order to allow the storing of old keys that may be used by some applications to decrypt old tickets during a transition.

When operating in FIPS mode, the keytab add command processing will pick FIPS compatible encryption types from the list of supported encryption types, to generate keys for the principal specified to add to the key table.

When operating in FIPS mode, the keys in source keytab are validated for FIPS compatibility. If the key is not FIPS compatible the key is ignored, and the processing continues with the merge operation. When the key is ignored, Kerberos message **EUVF06174E** is issued.

During a **keytab** add operation, An entry is created for each unique key type. Key table entries are not created for all supported encryption types because some of the encryption types share the same key type.

During a **keytab** merge operation, unsupported encryption types are not added. Entry fields are copied from the input keytab file to the target keytab file without change. If a keytab entry does not exist for a supported encryption type in the input keytab file, it cannot be created in the target keytab file.

When retrieving a key from the key table, the Kerberos runtime selects the appropriate key based upon the requested encryption type.

See “Encryption types and strong encryption” on page 9 and “Security runtime configuration profile” on page 37 for the supported key types and associated encryption types.

You must have a TGT in order to use the **keytab** check command. The KDC must be running and will be contacted to obtain service tickets during the check.

The **keytab** check attempts to obtain a service ticket for each principal in the keytab using the provided ticket-granting ticket (TGT), or for a given principal if specified on the keytab check command. It then uses an entry that matches each obtained ticket in principal name, version number, and encryption type to decrypt the ticket. Only entries that are used to decrypt service tickets are checked. These entries have the version number for a given principal that matches the highest version number for the same principal in the KDC, and usually have the same encryption type as used in the TGT. These entries are also used by service applications to decrypt tickets if the current setup is maintained. Only unsuccessful attempts are diagnosed. Entries for a principal that is missing in the keytab cannot be checked. If the setup is modified, it may be necessary to rerun the **keytab** check.

Examples

To add a key for principal **rwh** in the **/home/rwh/my_keytab** key table:

```
keytab add rwh -k /home/rwh/my_keytab
```

To list all of the entries in the **/home/rwh/my_keytab** key table:

```
keytab list -k /home/rwh/my_keytab
```

To merge all of the entries of keytab into my_keytab:

```
keytab merge keytab -k my_keytab
```

To merge all the entries only for principal fred in the preceding example (and ignore other entries):

```
keytab merge keytab fred -k my_keytab
```

kinit

Obtains or renews the Kerberos ticket-granting ticket.

Format

```
kinit [-s] [-c cache_name] [-T armor_ccache] [-k [-t keytab]] [-A]
              [-f] [-n] [-p] [-R] [-l end] [-r till] [ -X attribute[=value]] [principal]
```

Options

-s
Specifies that an initial ticket is to be obtained using the Kerberos principal associated with the current system identity. No password is used since the system has already verified the identity. The Kerberos security server must be running on the local system in order to use the -s option.

-r *time*
Specifies the renew time interval for a renewable ticket. The ticket may no longer be renewed after the expiration of this interval. The renew time must be greater than the end time. The ticket is not renewable if this option is not specified (a renewable ticket may still be generated if the requested ticket lifetime exceeds the maximum ticket lifetime). Refer to "Usage" on page 94 for more information on the time format.

-R
Specifies that an existing ticket is to be renewed. No other ticket options may be specified when renewing an existing ticket.

-p
Specifies that the ticket is to be proxiable. The ticket is not proxiable if this option is not specified.

-f
Specifies that the ticket is to be forwardable. The ticket is not forwardable if this option is not specified.

-n
Request a fully anonymous ticket which requires the KDC to be configured for PKINIT and that the client has enough PKINIT configuration to validate the KDC's certificate. Use a principal of the form @REALM (an empty principal name followed by an at-sign and the realm name) to request an anonymous ticket from a KDC that is not the default realm.

The kinit command fails if the -n option is specified with:

- the -X rsa_protocal=yes option - usage conflict message will be issued
- a principal name (other than the @<realm>principal) - usage conflict message will be issued
- a configuration that is not PKINIT capable - EUVF06171E will be issued

Note: If the Kerberos configuration indicates the use of the RSA protocol and the -n option is specified on the kinit command, the RSA protocol is disabled, and the Diffie-Hellman key agreement is used.

-A
Specifies that the ticket should not contain a list of client addresses. The ticket contains the local host address list if this option is not specified. When an initial ticket contains an address list, it can be used only from one of the addresses in the address list.

-l *time*
Specifies the ticket end time interval. The ticket may not be used after this interval has expired unless it has been renewed. The interval is set to 10 hours if this option is not specified. Refer to "Usage" on page 94 for more information on the time format.

-c *cache*
Specifies the name of the credentials cache that the kinit command uses. The default credentials cache is used if this option is not specified.

-T *armor_ccache*
Specifies that FAST pre-authentication is used and specifies the credential cache name that contains the armor ticket to be used. The armor credential cache must contain an anonymous PKINIT TGT for the kinit with FAST pre-authentication to succeed.

The kinit command fails if the -T option is specified with the -s, -k, or -t options. A usage conflict message will be issued.

-k
Specifies obtaining the key for the ticket principal from a key table. The user is prompted to enter the password for the ticket principal if this option, the **-R** option, or the **-s** option, is not specified.

-t *keytab*
Specifies the key table name. The default key table is used if this option is not specified and the **-k** option is specified. The **-t** option implies the **-k** option.

-X attribute=value
Specify a pre-authentication attribute and value to be passed. This option may be specified multiple times to specify multiple attributes. If -s is specified, this option will not take effect.For Public Key Cryptography for initial authentication (PKINIT), the attributes are:

- keyring - specify the key ring, key token or key database that contains the end entity certificate and its CA certificates in the format of <owner id>/<ring name> or token name in the format of *TOKEN*/<token name>, or key database name in the format of the full path name of the key database file. If a key database is used, its stash file needs to be specified too.
- stash- the stash file contains the password of the key database, in the format of <full path stash file name>, ignored if key database is not specified.
- rsa_protocol - specify 'yes' or 'no' to indicate whether to use RSA protocol, if no value is specified, it is defaulted to yes, i.e. RSA protocol is used; if this attribute is not specified or a value of no is specified, Diffie-Hellman protocol is used.

principal
Specifies the ticket principal. The principal is obtained from the credentials cache if the principal is not specified on the command line. If an anonymous ticket is being requested and the request is to be directed to a specific realm, use the @REALM principal name form (an empty principal name followed by an at-sign and the realm name). For example, to request an anonymous ticket from the MYCOMPANY.COM realm, specify: kinit -n @MYCOMPANY.COM.

Note: When using the kinit command to get an anonymous PKINIT ticket, the kinit command should be issued with the -n option, or the -n option and a principal name that only contains an "@" followed by the realm name.

Examples:

- To get an anonymous ticket from the default realm specified in the Kerberos configuration file:

```
kinit -n
```

- To get an anonymous ticket from the TESTSYS1.IBM.COM realm:

```
kinit -n @TESTSYS1.IBM.COM
```

The PKINIT configuration needed for obtaining an anonymous ticket are different than a PKINIT ticket. The requirement to have a client certificate and its issuers are lifted. The keyring, key database, or token only requires the certificates needed to verify the KDC's certificate used in the response.

To use FAST pre-authentication to obtain a TGT, it is necessary to obtain an anonymous PKINIT TGT to be used as the armor ticket, and then that armor ticket to request the TGT for the principal.

The armor ticket needs to be stored in a credential cache, which will be used by the FAST pre-authentication request.

Usage

The **kinit** command obtains or renews a Kerberos ticket-granting ticket. The KDC options specified by **kdc_default_options** in the Kerberos configuration file are used if no ticket options are specified on the **kinit** command.

If an existing ticket is not being renewed, the credentials cache is re-initialized and contains the new ticket-granting ticket received from the KDC. If the principal name is not specified on the command line and the **-s** option is not specified, the principal name is obtained from the credentials cache. The new credentials cache become the default credentials cache unless the cache name is specified using the **-c** option.

Ticket time values are expressed as *n***w***n***d**n**h***n***m***n***s** where *n* represents a number, **w** indicates weeks, **d** indicates days, **h** indicates hours, **m** indicates minutes, and **s** indicates seconds. The components must be specified in this order but any component may be omitted (for example, 4h5m represents four hours and 5 minutes and 1w2h represents 1 week and 2 hours). If only a number is specified, the default is hours.

Ticket time values that are actually obtained may not match those requested if limited by lifetime values setup in the KDC. Refer to “Configuring the primary security server for the realm” on page 22 for more details.

Examples

To obtain a ticket-granting ticket with a lifetime of ten hours that is renewable for one week:

```
kinit -l 10h -r 1w my_principal
```

To obtain an initial ticket based upon the current system identity:

```
kinit -s
```

To renew an existing ticket:

```
kinit -R
```

To obtain a ticket-granting ticket using PKINIT with the RSA encryption method based on the RACF key ring :

```
kinit –X keyring=KRBUSR/KRBRING -X rsa_protocol=yes my_principal
```

To obtain a ticket-granting ticket using PKINIT with the Diffie-Hellman encryption method based on the key database:

```
kinit –X keyring=/etc/skrb/clientgsk.kdb
      -X stash=/etc/skrb/clientgsk.sth
      -X rsa_protocol=no my_principal
```

To obtain a TGT using FAST pre-authentication.

Step 1: obtain an anonymous PKINIT TGT and store it in a credential cache file named armor.cache in the current directory.

```
kinit -n -c armor.cache
```

Note: If the kinit -n command is issued without the -c option, the anonymous PKINIT ticket will be stored in the default credential cache.

Step 2: Retrieve the armor ticket from the credential cache specified by the -T option, to obtain a TGT for the Gumby principal using FAST pre-authentication:

```
kinit -T armor.cache Gumby
```

- **klist:** add a new value 'a' for the anonymous ticket flag when displaying the ticket flags

klist

Displays the contents of a Kerberos credentials cache or key table.

Format

```
klist [-a] [-e] [-c] [-f] [-s] [-k] [-t] [-K] [filename]
```

Options

-a

Shows all tickets in the credentials cache, including expired tickets. Expired tickets are not listed if this option is not specified. This option is valid only when listing a credentials cache.

-e

Displays the encryption type for the session key and the ticket or the key table entry depending on what type of file is being processed. The information displayed for the session key and ticket will now display whether or not the key is FIPS compliant.

Example:

```
klist -e
  Ticket cache: FILE:/var/skrb/creds/krbcred_f5a38040
  Default principal: test1@ALPS4188.POK.IBM.COM

Server: krbtgt/ALPS4188.POK.IBM.COM@ALPS4188.POK.IBM.COM
  Valid 2020/09/10-10:28:20 to 2020/09/10-20:28:20
  Ticket encryption type: aes256-cts-hmac-sha384-192
  Session encryption type: des-cbc-crc : Non-FIPS Compliant
```

-c

Lists the tickets in a credentials cache. This is the default if neither the **-c** nor the **-k** option is specified. This option and the **-k** option are mutually exclusive.

-f

Shows the ticket flags using the following abbreviations. This option is valid only when listing a credentials cache.

- A - Preauthentication used
- a - Anonymous ticket
- C - Transited list checked by KDC
- D - Postdateable ticket
- d - Postdated ticket
- F - Forwardable ticket
- f - Forwarded ticket
- H - Hardware preauthentication used
- I - Initial ticket
- i - Invalid ticket
- P - Proxiable ticket
- p - Proxy ticket
- R - Renewable ticket

- O - Server can be a delegate

-s
Suppresses command output but sets the exit status to 0 if a valid ticket-granting ticket is found in the credentials cache. This option is valid only when listing a credentials cache.

-k
Lists the entries in a key table. This option and the **-c** option are mutually exclusive.

-t
Displays timestamps for key table entries. This option is valid only when listing a key table.

-K
Displays the encryption key value for each key table entry. This option is valid only when listing a key table.

filename
Specifies the name of the credentials cache or key table. The default credentials cache or key table is used if no filename is specified.

Usage

The **klist** command displays the contents of a Kerberos credentials cache or key table.

Examples

To list all of the entries in the default credentials cache:

```
klist
```

To list all of the entries in the **/krb5/my_keytab** key table with timestamps:

```
klist -k -t /krb5/my_keytab
```

```
klist -f
 Ticket cache: FILE:/var/skrb/creds/krbcred_b2199d80
 Default principal: WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS

Server: krbtgt/DCEIMGHG.PDL.POK.IBM.COM@DCEIMGHG.PDL.POK.IBM.COM
  Valid 2018/06/13-18:25:28 to 2018/06/14-04:25:28
  Flags: FIAa
```

kpasswd

Changes the password for a Kerberos principal.

Format

kpasswd [*principal*]

Options

principal
Specifies the principal whose password is to be changed. The principal is obtained from the default credentials cache if the principal is not specified on the command line.

Usage

The **kpasswd** command changes the password for the specified Kerberos principal using the password change service. You must supply the current password for the principal as well as the new password. The password change server applies any applicable password policy rules to the new password before changing the password.

You may not change the password for a ticket-granting service principal (**krbtgt/***realm*) using the **kpasswd** command.

If you have a SAF database, An RRSF local node must be defined in order to generate the corresponding Kerberos secret key whenever a password is changed. Refer to *z/OS Security Server RACF Security Administrator's Guide* for information on defining the local RRSF node.

kpropd

This is the Kerberos stand-alone database propagation catcher.

Format

```
kpropd [-r realm] [-P port] [-v]
```

Options

-P *port*
Specifies the port to use for the database propagation. The port assigned to the **krb5_prop** service is used if this option is omitted. Port 754 is used if the **krb5_prop** service is not defined.

-r *realm*
Specifies the database realm. The default realm obtained from the Kerberos configuration file is used if this option is omitted.

-v
The principal and policy names should be displayed as they are processed.

Usage

The **kpropd** command is used to receive a stand-alone database propagation from the primary KDC for the realm. The **kpropd** command is used when creating a secondary KDC or when recovering from a catastrophic database error.

To use the **kpropd** command for a secondary KDC, perform the following steps:

1. Stop the SKRBKDC started task if it is running on the secondary system.
2. Add the secondary system to the **/etc/skrb/home/kdc/kpropd.acl** configuration file on the primary system if it is not already defined. The propagation protocol should be set to **Manual** to prevent automatic propagation of database updates.
3. Use the **kdb5_ndbm destroy** command to remove an existing Kerberos database.
4. Use the **kdb5_ndbm stash** command to create the database master key stash file.
5. Create the **/etc/skrb/home/kdc/kpropd.acl** configuration file on the secondary system if it does not exist. The primary KDC for the realm must be listed in this file.
6. Use the **kadmin ktadd** command to create the **/var/skrb/krb5kdc/kpropd.ktf** key table if it does not exist. The principal name is **host/***host-name* where *host-name* is the primary host name for the local system. The primary host name is determined by doing a DNS lookup on the host name to get the IP address and then doing a DNS lookup on the IP address to get the host name.
7. Start the **kpropd** command on the secondary system.
8. Issue the **PROP** *host-name* console command on the primary KDC to initiate the database propagation.
9. Wait until the propagation is complete and the **kpropd** command ends.
10. Start the SKRBKDC started task on the secondary system.
11. Change the propagation protocol to **Update** in the **/etc/skrb/home/kdc/kpropd.acl** configuration file on the primary system.

ksetup

Manages Kerberos service entries in the LDAP directory for a Kerberos realm.

Format

```
ksetup [-h host-name] [-n bind-name] [-p bind-password] [-e]
```

Options

-h *host-name*
Specifies the host name for the LDAP server. The LDAP server specified in the Kerberos configuration file is used if this option is not specified.

-n *bind-name*
Specifies the distinguished name to use when binding to the LDAP server. The LDAP_BINDDN environment variable is used to obtain the name if this option is not specified.

-p *bind-password*
Specifies the password to use when binding to the LDAP server. The LDAP_BINDPW environment variable is used to obtain the password if this option is not specified.

-e
Echo each command line to **stdout**. This is useful when **stdout** is redirected to a file.

Usage

The **ksetup** command manages Kerberos service entries in the LDAP directory. The following subcommands are supported.

- **addadmin** *host-name:port-number realm-name*

  This subcommand adds an administration service entry for the specified realm. The port number is set to 749 if it is not specified. The fully-qualified host name should be used, so that it is resolved correctly no matter what default DNS name is in effect on the Kerberos clients. The default realm name is used if no realm name is specified.

- **addhost** *host-name realm-name*

  This subcommand adds a host entry for the specified realm. The fully-qualified host name should be used so that it is resolved correctly no matter what default DNS domain is in effect on the Kerberos clients. The default realm name is used if no realm name is specified. An error is displayed if the host entry already exists.

- **addkdc** *host-name:port-number realm-name*

  This subcommand adds a KDC entry for the specified realm. A host entry is created if one does not already exist. The port number is set to 88 if it is not specified. The fully-qualified host name should be used so that it is resolved correctly no matter what default DNS domain is in effect on the Kerberos clients. The default realm name is used if no realm name is specified. An existing KDC entry will be modified.

- **addpwd** *host-name:port-number real-name*

  This subcommand adds a password change service entry for the specified realm. The port number is set to 464 if it is not specified. The fully-qualified host name should be used so that it is resolved correctly no matter what default DNS domain is in effect on the Kerberos clients. The default realm name is used if no realm name is specified. An existing password service entry will be modified.

- **deladmin** *host-name real-name*

  This subcommand deletes an administration service entry for the specified host. The default realm name is used if no realm name is specified.

- **delhost** *host-name realm-name*

This subcommand deletes a host entry and any associated KDC specification from the specified realm. The default realm name is used if no realm name is specified.

- **delkdc** *host-name realm-name*

  This subcommand deletes a KDC entry for the specified host. The host entry itself is not deleted. The default realm name is used if no realm name is specified.

- **delpwd** *host-name realm-name*

  This subcommand deletes a password change service entry for the specified host. The default realm name is used if no realm name is specified.

- **listadmin** *realm-name*

  This subcommand lists the administration service entries for a realm. The default realm name is used if no realm name is specified.

- **listhost** *realm-name*

  This subcommand lists the host entries for a realm. The default realm name is used if no realm name is specified.

- **listkdc** *realm-name*

  This subcommand lists the KDC entries for a realm. The default realm name is used if no realm name is specified.

- **listpwd** *realm-name*

  This subcommand lists the password change service entries for a realm. The default realm name is used if no realm name is specified.

- **exit**

  This subcommand ends the **ksetup** command.

kvno

Displays the current key version number for a principal.

Format

kvno [*principal*]

Options

principal
Specifies the principal whose current key version number is to be displayed. The principal is obtained from the default credentials cache if the principal is not specified on the command line.

Chapter 6. Status codes

This chapter lists the status codes for z/OS Network Authentication Service. The status codes are listed in numerical order.

Major status values

GSS-API routines return GSS status codes as their function value. These codes indicate generic API errors and are common across GSS-API implementations. A GSS status code indicates a single API error from the routine and a single calling error. Additional status information can be contained in the GSS status code as supplementary information. The errors are encoded into a 32-bit GSS status code as follows:

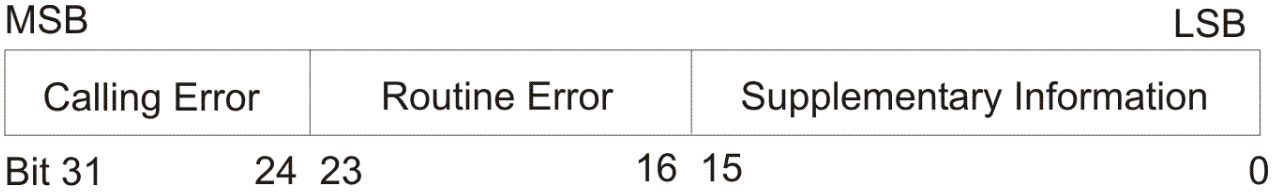

Figure 3. GSS status code bit locations

If a GSS-API routine returns a GSS status code whose upper 16 bits contain a nonzero value, the call failed. If the calling error field is nonzero, the application's call of the routine was in error. In addition, the routine can indicate additional information by setting one or more bits in the supplementary information field of the status code.

The following table lists the GSS-API calling errors and their meanings:

Table 10. GSS-API calling errors

Error (Error Number, Hexidecimal)	Meaning
GSS_S_CALL_INACCESSIBLE_READ (01)	Unable to read an input parameter
GSS_S_CALL_INACCESSIBLE_WRITE (02)	Unable to write an output parameter
GSS_S_CALL_BAD_STRUCTURE (03)	Incorrect parameter structure

The following table lists the GSS-API routine errors and their meanings:

Table 11. GSS-API routine errors

Error (Error Number, Hexidecimal)	Meaning
GSS_S_BAD_MECH (01)	Mechanism is not supported
GSS_S_BAD_NAME (02)	Name is not valid
GSS_S_BAD_NAMETYPE (03)	Name type is not valid
GSS_S_BAD_BINDINGS (04)	Channel bindings are not correct
GSS_S_BAD_STATUS (05)	Status value is not valid
GSS_S_BAD_SIG (06)	Token signature is not correct
GSS_S_NO_CRED (07)	No credentials supplied
GSS_S_NO_CONTEXT (08)	No context established
GSS_S_DEFECTIVE_TOKEN (09)	Token is not valid
GSS_S_DEFECTIVE_CREDENTIAL (0A)	Credential is not valid

Table 11. GSS-API routine errors (continued)	
Error (Error Number, Hexidecimal)	**Meaning**
GSS_S_CREDENTIALS_EXPIRED (0B)	Credentials have expired
GSS_S_CONTEXT_EXPIRED (0C)	Context has expired
GSS_S_FAILURE (0D)	Routine failed (check minor status)
GSS_S_BAD_QOP (0E)	Bad quality-of-protection value
GSS_S_UNAUTHORIZED (0F)	Operation not authorized by local security policy
GSS_S_UNAVAILABLE (10)	Operation or option not available
GSS_S_DUPLICATE_ELEMENT (11)	Credential element already exists
GSS_S_NAME_NOT_MN (12)	Not a mechanism name

The following table lists the GSS-API supplementary status bits and their meanings:

Table 12. GSS-API supplementary status bits	
Status Bit (Number, Hexidecimal)	**Meaning**
GSS_S_CONTINUE_NEEDED (0001)	Call routine again to complete request
GSS_S_DUPLICATE_TOKEN (0002)	Token is duplicate of earlier token
GSS_S_OLD_TOKEN (0004)	Token validity period has expired
GSS_S_UNSEQ_TOKEN (0008)	Later token has already been processed
GSS_S_GAP_TOKEN (0010)	Skipped predecessor token detected

Kerberos administration database (numbers 01B79C00 - 01B79CFF)

01B79C01 Principal or policy already exists.

Explanation

The principal or policy entry already exists in the Kerberos database.

User response

Specify a name that does not already exist.

01B79C02 Principal or policy does not exist.

Explanation

The principal or policy entry does not exist in the Kerberos database.

User response

Specify a name that does exist.

01B79C03 Database is not initialized.

Explanation

The Kerberos database is not initialized.

User response

Report the problem to the owner of the Kerberos administration server.

01B79C04 Policy name is not valid.

Explanation

The policy name is not valid.

User response

Contact the owner of the Kerberos administration server to obtain the policy name guidelines for that server.

01B79C05 Principal name is not valid.

Explanation

The principal name is not valid.

User response

Contact the owner of the Kerberos administration server to obtain the principal name guidelines for that server.

01B79C06 Database inconsistency detected.

Explanation

A database inconsistency has been detected by the Kerberos administration server.

User response

Report the problem to the owner of the Kerberos administration server.

01B79C07 XDR encoding error.

Explanation

The Kerberos administration server detected an error while decoding the administration request or while encoding the administration response.

User response

Report the problem to the owner of the Kerberos administration server.

01B79C08 Database operation failed.

Explanation

The Kerberos administration server is unable to complete a database operation.

User response

Report the problem to the owner of the Kerberos administration server.

01B79C09 Database lock mode is not valid.

Explanation

The Kerberos administration server is unable to lock the database.

User response

Report the problem to the owner of the Kerberos administration server.

01B79C0A Unable to lock database.

Explanation

The Kerberos administration server is unable to lock the database.

User response

Report the problem to the owner of the Kerberos administration server.

01B79C0B Database is not locked.

Explanation

The Kerberos administration server is unable to perform a database operation.

User response

Report the problem to the owner of the Kerberos administration server.

01B79C0C Administration database lock file is missing.

Explanation

The Kerberos administration server is unable to perform a database operation.

User response

Report the problem to the owner of the Kerberos administration server.

01B79C0D Insufficient permission to lock file.

Explanation

The Kerberos administration server is unable to perform a database operation.

User response

Report the problem to the owner of the Kerberos administration server.

GSS-API Kerberos mechanism codes (numbers 025EA100 - 025EA1FF)

025EA100 **Principal is not found in credentials cache.**

Explanation

The principal name in the default credentials cache is not the same as the principal name on the GSS-API request.

User response

Either specify the correct principal or specify GSS_C_NO_NAME to use the default principal.

025EA101 **Principal is not found in key table.**

Explanation

The principal specified on a GSS-API request is not found in the key table.

User response

Specify a principal that is defined in the key table or use the KRB5_KTNAME environment variable to specify a different key table.

025EA102 **Ticket-granting ticket is not found in credentials cache.**

Explanation

GSS-API is creating a credential that can be used to initiate a security context. The credentials cache does not contain a ticket-granting ticket that can be used to obtain service tickets.

User response

Obtain a valid ticket-granting ticket for the principal to be used to initiate the security context. Then retry the request.

025EA104 **Context already established.**

Explanation

The **gss_init_sec_context()** function is called to continue setting up the security context. However, the security context has already been established. The application should not call **gss_init_sec_context()** again unless the previous call returned a major status of GSS_C_CONTINUE_NEEDED.

User response

Verify that the previous call to **gss_init_sec_context()** returned a major status of GSS_C_CONTINUE_NEEDED.

025EA105 **Signature algorithm is not supported.**

Explanation

The signature algorithm is not supported by the local system or is not compatible with the current security context.

User response

Ensure that the selected signature algorithm is supported by both the local and the remote system and is compatible with the encryption key associated with the current security context. Contact your service representative if the error persists.

025EA106 **Length value is not correct.**

Explanation

The length of a field in an input token is not correct.

User response

Ensure that the input token is not modified. Contact your service representative if the error persists.

025EA107 **Context is not established.**

Explanation

A GSS-API function was called that requires an established security context. The supplied context has been initiated but the response from the context acceptor has not been received.

User response

Process the response token before attempting to use the security context.

025EA108 **Context identifier is not valid.**

Explanation

An unassigned context identifier is specified on a GSS-API function call.

User response

Specify a valid context identifier and then retry the request.

025EA109 Credential identifier is not valid.

Explanation

An unassigned credential identifier is specified on a GSS-API function call.

User response

Specify a valid credential identifier and then retry the request.

025EA10B Token sequence number is not valid.

Explanation

The token sequence number is not correctly formed. This error can occur if the token is modified or if the session key in the security context is not correct.

User response

Ensure that the token is not modified and that the correct security context is used to process the token.

025EA140 Token pad characters are not valid.

Explanation

The token does not contain the correct pad characters.

User response

Verify that the token is not modified. Contact your service representative if the error persists.

025EA141 Data privacy service is not available.

Explanation

The **gss_unwrap_()** function was called to process a message that was encrypted by the sender. However, data encryption support is not available on the local system.

User response

Do not request message confidentiality protection unless both systems provide data encryption services.

025EA142 Seal algorithm is not supported.

Explanation

The seal algorithm is not supported by the local system or is not compatible with the current security context.

User response

Ensure that both systems are at compatible software levels. Contact your service representative if the error persists.

025EA143 Token length is not correct.

Explanation

The length of the buffer containing the input token is not correct.

User response

Verify that the correct length is specified for the token buffer. Contact your service representative if the error persists.

025EA144 Encryption type is not supported.

Explanation

The encryption type specified in the input token is not valid or is not supported by the current software level.

User response

Ensure that both systems are at compatible software levels. Contact your service representative if the error persists.

025EA145 No key is available to accept a security context.

Explanation

The **gss_accept_sec_context()** function failed because it is unable to obtain the session key for the security context.

User response

Contact your service representative if the error persists.

025EA146 Bindings in token do not match supplied bindings.

Explanation

The channel bindings specified on the **gss_accept_sec_context()** call do not match the channel bindings contained in the input token.

User response

Ensure that the input token is not modified and that the context initiator is specifying the correct channel bindings on the **gss_init_sec_context()** call.

025EA147 **Checksum in token is not valid.**

Explanation

The checksum in the input token does not have the correct type and length values.

User response

Ensure that both systems are at compatible software levels. Contact your service representative if the error persists.

025EA148 **Context is not in the correct state.**

Explanation

The context state is not valid for the requested operation. This error can occur if an operation is attempted while the context is still in the initialization state. This error can also occur if **gss_init_sec_context()** or **gss_accept_sec_context()** is called after the context has been established.

User response

Ensure that the context is in the correct state for the requested operation.

025EA149 **Locking error is detected.**

Explanation

An internal locking error is detected.

User response

Contact your service representative if the error persists.

025EA14A **No mechanism credentials available.**

Explanation

The credential context supplied on the **gss_init_sec_context**or the **gss_inquire_cred_by_mech** API function does not contain credentials for the requested mechanism.

User response

Specify a credential context that contains the necessary mechanism credentials. The **gss_acquire_cred** API function can be used to create the credential context.

025EA14B **No internal name provided for requested mechanism.**

Explanation

A *gss_name_t* parameter does not contain an internal representation that is valid for the requested mechanism. Names returned by one mechanism may not be used with a different mechanism. The **gss_import_name** API function can be used to generate a *gss_name_t* that contains internal representations for all of the supported mechanisms.

User response

Provide a *gss_name_t* that is valid for the requested mechanism.

GSS-API LIPKEY/SPKM mechanism codes (numbers 025EA160-025EA18F)

025EA160 **Certificate management services are not available.**

Explanation

The certificate management services library cannot be loaded. The library name is GSKCMS31 for 31-bit applications or GSKCMS64 for 64-bit applications.

User response

Ensure that the required load module is installed and accessible. Then retry the failing operation. Contact your service representative if the error persists.

025EA161 **Unable to decode X.509 certificate.**

Explanation

Certificate management services is unable to decode an X.509 certificate. Additional information for the error can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Ensure that the key database or SAF key ring has not been modified. Contact your service representative if the error persists.

025EA162 Unable to decode X.509 certificate revocation list.

Explanation

Certificate management services is unable to decode an X.509 certificate revocation list (CRL). Additional information for the error error can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Ensure that the LDAP entry for the certification authority contains a valid CRL. Contact your service representative if the error persists.

025EA163 Unable to convert a distinguished name to an X.509 name.

Explanation

Certificate management services is unable to convert a distinguished name (DN) string to an X.509 name. Additional information for the error error can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Ensure that a valid distinguished name is specified. Refer to RFC 2253 (UTF-8 String Representation of Distinguished Names) for more information on the string representation of a distinguished name. Contact your service representative if the error persists.

025EA164 Unable to convert an X.509 name to a distinguished name.

Explanation

Certificate management services is unable to convert an X.509 name to a distinguished name (DN) string. Additional information for the error error can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Ensure that the X.509 certificate or certificate revocation list has not been modified. Contact your service representative if the error persists.

025EA165 Unable to decode X.509 name.

Explanation

Certificate management services is unable to decode an X.509 name. Additional information for the error error can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Ensure that the key database or SAF key ring has not been modified. Contact your service representative if the error persists.

025EA166 Unable to encode X.509 object.

Explanation

Certificate management services is unable to encode an X.509 object. Additional information for the error error can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Contact your service representative if the error persists.

025EA167 Unable to obtain certificate.

Explanation

Certificate management services is unable to read a certificate from the key database or SAF key ring. Additional information for the error error can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Ensure that the key database or SAF key ring has not been modified and that the application has access. Contact your service representative if the error persists.

025EA168 Certificate is not found.

Explanation

The requested X.509 certificate is not found in the key database or SAF key ring.

User response

Specify a valid certificate name.

025EA169 Certificate is not usable.

Explanation

The requested X.509 certificate is not usable. This error can occur if the certificate is expired or is not marked as trusted. This error can also occur for an application certificate if there is no private key for the certificate.

User response

Verify that the certificate is not expired and is marked as trusted. Verify that an application certificate has a private key in the key database or SAF key ring.

025EA16A More than one matching certificate.

Explanation

The key database or SAF key ring contains multiple certificates matching the supplied subject name.

User response

Either remove the extra certificates or specify a unique certificate subject name.

025EA16B Certificate database is not open.

Explanation

The key database or SAF key ring is not open. This error can occur if the GSS_KEYRING_NAME environment variable is not defined or if an error occurred when the database was opened.

User response

Ensure that the GSS_KEYRING_NAME environment variable is defined. In addition, either GSS_KEYRING_PW or GSS_KEYRING_STASH must be defined if you are using a key database instead of a SAF key ring.

025EA16C Target name is not valid.

Explanation

The target name for the acceptor of a GSS-API security context cannot be validated. This indicates the target name specified by the security context initiator cannot be used with the X.509 certificate supplied by the acceptor.

The target certificate must satisfy one of the following target name criteria:

- The target name must match the subject name or the common name component of the subject name.
- The target name must match a DN entry or the common name component of a DN entry for the subject alternate name in the target certificate.
- The host name portion of a target name created from a service name must match the common name component of the subject name.
- The host name portion of a target name created from a service name must match a DNS entry for the subject alternate name in the target certificate.

User response

Ensure that the target certificate is valid and the target name can be used with the certificate.

025EA16D Certificate is not valid.

Explanation

The X.509 certificate supplied by the communication partner cannot be validated. This error can occur if the certificate is not valid or has expired.

User response

Ensure that the supplied certificate is valid and has not expired.

025EA16E Certificate is not trusted.

Explanation

The X.509 certificate supplied by the communication partner is not trusted. This error can occur if the certificate has been revoked or a required certification authority root certificate is not found in the local database.

User response

Ensure that the certificate has not been revoked and that the local database contains the necessary certification authority root certificate.

025EA16F Unexpected certificate management error.

Explanation

An unexpected error is returned by Certificate Management Services. Additional information can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Refer to the System SSL documentation for a description of the CMS error code.

025EA170 Incomplete certification chain.

Explanation

The certification chain for the application certificate is incomplete. The key database or SAF key ring must contain all certificates in the certification chain up to

and including the root certificate for the certification authority.

User response

Ensure that all certificates in the certification chain are in the key database or SAF key ring and are trusted and not expired.

025EA171 Unable to generate Diffie-Hellman key pair.

Explanation

A Diffie-Hellman public/private key pair cannot be generated. Additional information can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Refer to the System SSL documentation for a description of the CMS error code.

025EA172 Unable to generate Diffie-Hellman secret.

Explanation

The Diffie-Hellman shared secret cannot be generated. Additional information can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Refer to the System SSL documentation for a description of the CMS error code.

025EA173 Diffie-Hellman parameters are not valid.

Explanation

The Diffie-Hellman group parameters are not valid. Additional information can be obtained by turning on debug messages for subcomponent KRB-GSSAPI.

User response

Refer to the System SSL documentation for a description of the CMS error code.

025EA174 Unable to sign token data.

Explanation

The GSS-API token data cannot be signed using the private key obtained from the X.509 certificate. Additional information can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Refer to the System SSL documentation for a description of the CMS error code.

025EA175 Unable to verify token data signature.

Explanation

The signature for a GSS-API token cannot be verified using the public key obtained from the X.509 certificate. Additional information can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Refer to the System SSL documentation for a description of the CMS error code.

025EA176 Token signature is not correct.

Explanation

The signature for a GSS-API token is not correct. This error can occur if the token data has been modified or the wrong X.509 certificate is used to verify the signature.

User response

Verify that the GSS-API token data is not modified. Contact your service representative if the error persists.

025EA177 Protocol version is not supported.

Explanation

The SPKM request token does not specify a protocol version supported by the local system. The only supported version is SPKM protocol version 0 as documented in RFC 2025 (Simple Public Key GSS-API Mechanism).

User response

Ensure that the latest software level is installed.

025EA178 Protocol error is detected.

Explanation

An SPKM token violates the protocol as defined in RFC 2025 (Simple Public Key GSS-API Mechanism).

User response

Contact your service representative.

025EA179 Target name is incorrect.

Explanation

An SPKM context cannot be accepted because the target name in the context cannot be used with the X.509 certificate in the acceptor credential.

User response

Change the application to specify the correct target name or use a different X.509 certificate with the target application.

025EA17A Bindings in token do not match supplied bindings.

Explanation

The channel bindings specified on the gss_accept_sec_context() call do not match the channel bindings contained in the input token.

User response

Ensure that the input token is not modified and that the context initiator is specifying the correct channel bindings on the gss_init_sec_context() call.

025EA17B Algorithm is not supported.

Explanation

An unsupported algorithm is specified in a GSS-API token.

User response

Verify that the latest software level is installed. Contact your service representative if the problem persists.

025EA17C Diffie-Hellman modulus is too small.

Explanation

The SPKM-3 and LIPKEY security mechanisms use the Diffie-Hellman key agreement algorithm to construct a shared secret. The Diffie-Hellman parameters provided by the context initiator cannot be used because the modulus is not large enough to construct key material for the negotiated algorithm set.

User response

Contact your service representative.

025EA17D Unable to decode private key.

Explanation

Certificate management services is unable to decode a private key. Additional information for the error error can be obtained by turning on debug messages for subcomponent KRB_GSSAPI.

User response

Ensure that the key database or SAF key ring has not been modified. Contact your service representative if the error persists.

025EA17E Password prompt canceled by user.

Explanation

The user canceled the password prompt issued by the GSS-API runtime. As a result, an initiate credential could not be obtained for use with the LIPKEY security mechanism.

User response

Enter the password when prompted or select a different security mechanism.

025EA17F User authentication rejected by target.

Explanation

GSS-API is unable to establish a security context because the __passwd function used to validate the user's password indicates a failure. This could be because :

- The supplied user name and password are not valid.
- There is a problem with the setup of the program controlled environment.

User response

Examine the console log to determine if it is a problem with the program controlled environment.

- For password issues, refer to *z/OS UNIX System Services Programming: Assembler Callable Services Reference*.
- For program controlled environment issues, refer to *z/OS UNIX System Services Planning*.

Kerberos administration codes (numbers 029C2500 - 029C25FF)

029C2500 Operation failed.

Explanation

A Kerberos administration request has failed. There may be additional information on the cause of the failure in the administration server log.

User response

Correct the cause of the failure and retry the request. Contact your service representative if the error persists.

029C2501 Operation requires 'get' privilege.

Explanation

A Kerberos administration request cannot be processed because the client is not authorized to retrieve an entry from the security registery.

User response

Repeat the request using a client with the proper authorization.

029C2502 Operation requires 'add' privilege.

Explanation

A Kerberos administration request cannot be processed because the client is not authorized to add an entry to the security registry.

User response

Repeat the request using a client with the proper authorization.

029C2503 Operation requires 'modify' privilege.

Explanation

A Kerberos administration request cannot be processed because the client is not authorized to modify an entry in the security registry.

User response

Repeat the request using a client with the proper authorization.

029C2504 Operation requires 'delete' privilege.

Explanation

A Kerberos administration request cannot be processed because the client is not authorized to delete an entry from the security registry.

User response

Repeat the request using a client with the proper authorization.

029C2505 Insufficient authorization for operation.

Explanation

A Kerberos administration request cannot be processed because the client does not have the necessary authorization.

User response

Repeat the request using a client with the proper authorization.

029C2506 Database inconsistency detected.

Explanation

The Kerberos administration server detects an error in the security registry.

User response

Contact your service representative.

029C2507 Principal or policy already exists.

Explanation

A Kerberos administration request cannot be processed because the principal or policy already exists in the security registry.

User response

Choose a different name for the new registry object or delete the existing registry object.

029C2508 Communication failure with administration server.

Explanation

A Kerberos administration request cannot be processed due to an RPC communication failure.

User response

Verify that the administration server is running and there are no network problems. Contact your service representative if the error persists.

029C2509 **No administration server available.**

Explanation

No Kerberos administration server is available for the requested realm. This error can occur if no administrator server is defined or a session cannot be established with the administration server.

User response

Verify that an administration server is defined for the realm and the server is running.

029C250A **Key version mismatch for password history principal.**

Explanation

The key version number for the password history principal is incorrect.

User response

Verify that the key table contains the correct key for the principal. Contact your service representative if the error persists.

029C250B **Administration server connection is not initialized.**

Explanation

A connection to the Kerberos administration server has not been initialized. Use the **kadm5_init_with_password()**, **kadm5_init_with_skey()**, or **kadm5_init_with_creds()** routine to initialize a connection.

User response

Correct the application to initialize the connection before calling any administration routines.

029C250C **Principal does not exist.**

Explanation

The requested principal does not exist in the security registry.

User response

None

029C250D **Policy does not exist.**

Explanation

The requested policy does not exist in the security registry.

User response

None

029C250E **Field mask is not valid for operation.**

Explanation

The configuration parameters field mask is not valid for the requested administration operation. Refer to the API documentation to determine which field mask settings are valid.

User response

Correct the application to specify a valid field mask.

029C250F **Character class count is not valid.**

Explanation

The number of character classes is not valid for the requested administration operation.

User response

Contact the owner of the Kerberos administration server to determine the allowable range for the number of character classes.

029C2510 **Password length is not valid.**

Explanation

The password length is not valid. Contact your security administration to get the password requirements for your installation.

User response

Contact the owner of the Kerberos administration server to determine the allowable range for the password length.

029C2511 **Policy name is not valid.**

Explanation

The policy name is not valid.

User response

Specify a valid policy name.

029C2512 Principal name is not valid.

Explanation

The principal name is not valid.

User response

Specify a valid principal name.

029C2513 Auxillary attributes are not valid.

Explanation

Auxillary attributes are not valid.

User response

Specify a valid set of auxillary attributes.

029C2514 Password history count is not valid.

Explanation

The password history count is not valid.

User response

Contact the owner of the Kerberos administration server to determine the allowable range for the password history count.

029C2515 Minimum password lifetime is not valid.

Explanation

The minimum password lifetime is not valid. The minimum lifetime must not be greater than the maximum lifetime.

User response

Specify a valid password lifetime.

029C2516 Password is too short.

Explanation

The password is too short. Contact your security administrator to get the password requirements for your installation.

User response

Specify a valid password.

029C2517 Password does not contain enough character classes.

Explanation

The password does not contain characters from enough different character classes. Contact your security administrator to get the password requirements for your installation.

User response

Specify a valid password.

029C2518 Password is in the password dictionary.

Explanation

The password is in the password dictionary. Contact your security administrator to get the password requirements for your installation.

User response

Specify a valid password.

029C2519 Password cannot be reused.

Explanation

The password has already been used and cannot be used again. Contact your security administrator to get the password requirements for your installation.

User response

Specify a valid password.

029C251A Password minimum lifetime has not expired.

Explanation

The password cannot be changed until the minimum lifetime has expired. Contact your security administrator to get the password requirements for your installation.

User response

Wait until the minimum lifetime has elapsed before attempting to change the password.

029C251B Policy is in use.

Explanation

The policy cannot be deleted because it is referenced by one or more principals.

User response

Delete the principals that reference the policy before deleting the policy.

029C251C Connection to server is already initialized.

Explanation

The connection to the administration server is already initialized.

User response

None

029C251D Password is not correct.

Explanation

The password entered is not the correct one for the current client.

User response

Specify the correct password.

029C251E Protected principal cannot be modified.

Explanation

The principal is protected and cannot be modified.

User response

None

029C251F Administration server handle is not valid.

Explanation

The server handle passed to an administration function is not valid.

User response

Correct the application to pass the correct service handle to the administration function.

029C2520 Structure version is not valid.

Explanation

The structure version number is not valid.

User response

Change the application to specify a valid version number.

029C2521 Old structure version is not supported.

Explanation

An obsolete structure version number was specified by the application.

User response

Change the application to use the current structure version.

029C2522 New structure version is not supported.

Explanation

The structure version is not supported by the current level of the administration support.

User response

Change the application to use a supported structure version.

029C2523 API version is not valid.

Explanation

The API version is not valid.

User response

Change the application to specify a valid version number.

029C2524 Old API version is not supprted by the administration library.

Explanation

An obsolete API version is specified by the application.

User response

Change the application to use the current API version.

029C2525 **Old API version is not supprted by the administration server.**

Explanation

An obsolete API version is specified by the application.

User response

Change the application to use the current API version.

029C2526 **New API version is not supprted by the administration library.**

Explanation

The API version is not supported by the current level of the administration support.

User response

Change the application to use a supported API version.

029C2527 **New API version is not supported by the administration server.**

Explanation

The API version is not supported by the current level of the administration support.

User response

Change the application to use a supported API version.

029C2528 **Required administration principal is not found.**

Explanation

A required administration principal was not found in the security registry.

User response

Contact your service representative.

029C2529 **Principal cannot be renamed.**

Explanation

The salt type for the principal does not allow the principal to be renamed.

User response

None

029C252A **Administration client configuration parameters are not valid.**

Explanation

The client configuration parameters are not valid.

User response

Verify that the application parameters are valid and that valid parameters are specified in the Kerberos profile. Contact your service representative if the error persists.

029C252B **Administration server configuration parameters are not valid.**

Explanation

The server configuration parameters are not valid.

User response

Verify that valid parameters are specified in the server configuration profile. Contact your service representative if the errror persists.

029C252C **Operation requires 'list' privilege.**

Explanation

A Kerberos administration request cannot be processed because the client is not authorized to list the contents of the security registry.

User response

Repeat the rquest using a client with the proper authorization.

029C252D **Operation requires 'change-password' privilege.**

Explanation

A Kerberos administration request cannot be processed because the client is not authorized to change the password for a principal.

User response

Repeat the rquest using a client with the proper authorization.

029C252E **GSS-API error.**

Explanation

A Kerberos administration request cannot be processed due to an error reported by GSS-API function.

User response

Contact your service representative if the error persists.

029C252F **Tagged data list type is not valid.**

Explanation

A tagged data list type contains a type code that is not valid.

User response

Contact your service representative if the error persists.

029C2530 **Required configuration parameter is missing.**

Explanation

A required configuration parameter is not specified in the Kerberos configuration profile.

User response

Verify that all administration server configuration parameters are specified. Contact your service representative if the error persists.

029C2531 **Administration server host name is not valid.**

Explanation

The administration server host name is not valid.

User response

Specify a valid host name.

029C2532 **Operation requires 'set-key' privilege.**

Explanation

A Kerberos administration request cannot be processed because the client is not authorized to set the encryption key for a principal.

User response

Repeat the request using a client with the proper authorization.

029C2533 **Duplicate encryption types specified.**

Explanation

Duplicate encryption types specified when setting the encryption key for a principal. This error can also occur if two encryption types are specified that use the same encryption key (for example, ENCTYPE_DES_CBC_CRC and ENCTYPE_DES_CBC_MD5 use the same DES encryption key).

User response

Do not specify duplicate encryption types.

029C2535 **Encryption type mismatch.**

Explanation

The key-salt entries do not match the corresponding key entries on a call to the **kadm5_setkey_principal_3()** routine.

User response

Specify a matching key-salt entry for each key entry.

029C25F0 **Too many matching database records.**

Explanation

A database search request resulted in more than 1000 matching records.

User response

Repeat the request using a more restrictive search expression.

029C25F1 **Database record is too big.**

Explanation

An attempt to create a Kerberos database entry failed because the record is too big. The Kerberos database has a maximum record size of 1024 bytes. This size includes the record key and any database overhead.

User response

Reduce the amount of data stored for the failing principal or policy.

029C25F2 **Password change rejected.**

Explanation

The password change was rejected by the system authorization facility. This error occurs if the password is too long or violates the password policy defined for the system.

User response

Select a different password. Contact your system administrator if the error persists.

029C25F3 **Unsupported encryption type.**

Explanation

The encryption type is not supported by the current software level.

User response

Upgrade to a software level that supports the encryption type.

029C25F4 **Unsupported salt type.**

Explanation

The password salt type is not supported by the current software level.

User response

Upgrade to a software level that supports the salt type.

029C25F5 **Function not supported.**

Explanation

The requested function is not supported by the Kerberos administration server.

User response

Refer to the documentation for the Kerberos administration server to determine which administration functions are supported.

029C25F6 **Function disabled.**

Explanation

The requested function is currently disabled by the Kerberos administration server.

User response

Retry the request when Kerberos administration services are enabled. Contact your Kerberos administrator if the error persists.

029C25F7 **Target is not a Kerberos object.**

Explanation

A request to rename or delete a registry object cannot be performed because the object was not created by the Kerberos administration service.

User response

The registry object must be renamed or deleted with the same service that was used to create it.

ASN.1 operations codes (numbers 6EDA3600 - 6EDA36FF)

6EDA3600 **ASN.1 is unable to obtain the system time.**

Explanation

An ASN.1 encode/decode function is unable to obtain the current system time. This error can occur if the time provider is not running.

User response

Verify that the time provider is running and is configured properly. Then retry the request. Contact your service representative if the error persists.

6EDA3601 **An ASN.1 structure is missing a required field.**

Explanation

An ASN.1 encode function was unable to process a request because an input structure is missing a required field. This error can also occur on a decode request if the byte stream was created by a different level of the ASN.1 software.

User response

Verify that all input structures contain all required fields and then retry the request. Contact your service representative if the error persists.

6EDA3602 **ASN.1 encounters an unexpected field number.**

Explanation

An ASN.1 decode function is unable to process a request because the input byte stream contains a misplaced field. This error can occur if the byte stream was created by a different level of the ASN.1 software.

User response

Verify that the input byte stream has not been modified and then retry the request. Contact your service representative if the error persisits.

6EDA3603 ASN.1 type number is not correct.

Explanation

An ASN.1 decode function is unable to process a request because the input byte stream contains an invalid type specification. This error can occur if the byte stream was created by a different level of the ASN.1 software.

User response

Verify that the input byte stream has not been modified and then retry the request. Contact your service representative if the error persists.

6EDA3604 ASN.1 value is too large.

Explanation

An ASN.1 encode/decode function is unable to process a request because a data value is too large.

User response

Verify that all data values are within the defined limits for that data type. Contact your service representative if the error persists.

6EDA3605 ASN.1 endecoding operation fails at end of data.

Explanation

An ASN.1 encoding function was unable to process a request because the end of the encoded stream was reached prematurely.

User response

Verify that the encoded stream has not been modified and then retry the request. Contact your service representative if the error persists.

6EDA3606 ASN.1 identifier does not match expected value.

Explanation

An ASN.1 decode function was unable to process a request because an internal identifier does not match the expected value for the identifier. This error can occur if the byte stream was created by a different level of the ASN.1 software.

User response

Verify that the input byte stream has not been modified and then retry the request. Contact your service representative if the error persists.

6EDA3607 ASN.1 length is not correct.

Explanation

An ASN.1 encode/decode function was unable to process a request because the length of a field does not match the expected value.

User response

Verify that all field lengths are correct and then retry the request. Contact your service representative if the error persists.

6EDA3608 ASN.1 encoded byte stream is not valid.

Explanation

An ASN.1 decode function was unable to process a request because the input byte stream is formatted incorrectly. This error can occur if the byte stream was created by a different level of the ASN.1 software.

User response

Verify that the input byte stream has not been modified and then retry the request. Contact your service representative if the error persists.

6EDA3609 ASN.1 is unable to parse the request.

Explanation

An ASN.1 decode function was unable to process a request because the input byte stream cannot be parsed. This error can occur if the byte stream was created by a different level of the ASN.1 software.

User response

Verify that the input byte stream has not been modified and then retry the request. Contact your service representative if the error persists.

6EDA360A **ASN.1 object identifier element count is not valid.**

Explanation

An object identifier must have at least three elements.

User response

Correct the application to provide a valid object identifier.

6EDA360B **First object identifier element value is not valid.**

Explanation

The first element of an object identifier must be 0, 1, or 2.

User response

Correct the application to provide a valid object identifier.

6EDA360C **Second object identifier element value is not valid.**

Explanation

The second element of an object identifier must be between 0 and 39 when the first element is 0, or between 40 and 79 if the first element is 1.

User response

Correct the application to provide a valid object identifier.

GSS-API codes (numbers 861B6D00 - 861B6DFF)

861B6D00 **Service name is not valid.**

Explanation

The supplied name is not a valid Kerberos service name.

User response

Specify a valid service name.

861B6D01 **UID string is not valid.**

Explanation

The supplied UID string is not valid

User response

Specify a valid UID string

861B6D02 **UID does not resolve to a user.**

Explanation

The specified UID does not resolve to a valid user on the local system.

User response

Specify a UID that is valid on the local system and then retry the request.

861B6D03 **Control block validation fails.**

Explanation

A GSS-API control block or token is not correct. This error can occur if an exported context or credential is imported by an earlier version of the Kerberos runtime.

User response

Upgrade the Kerberos runtime to the same level on all systems.

861B6D04 **Unable to allocate memory.**

Explanation

A GSS-API operation is unable to allocate memory.

User response

Increase the memory available to the application and then retry the request. Contact your service representative if the error persists.

861B6D05 **Message context is not valid.**

Explanation

The **gss_display_status()** routine was called with an incorrect message context.

User response

Initialize the message context to zero before the first call to the **gss_display_status()** routine.

861B6D06 **Buffer length is not correct.**

Explanation

The length of the supplied buffer is not correct for the operation being attempted.

User response

Provide a buffer of the proper length and then retry the request.

861B6D07 Credential usage type is not valid.

Explanation

The credential usage must be GSS_C_INITIATE, GSS_C_ACCEPT, or GSS_C_BOTH when acquiring a credential. The credential usage must be GSS_C_INITIATE or GSS_C_BOTH when initiating a security context. The credential usage must be GSS_C_ACCEPT or GSS_C_BOTH when accepting a security context.

User response

Specify a credential usage that is valid for the operation being attempted.

861B6D08 Quality of protection is not valid.

Explanation

The quality of protection (QOP) value specified for the GSS-API operation is not valid or is not supported by the current software level.

User response

Specify a valid quality of protection value and then retry the request.

861B6D0A Security mechanism is not correct.

Explanation

The security mechanism specified in the token header is not correct.

User response

Verify that the token was created using a security mechanism that is supported by the current software level and that this security mechanism is the same security mechanism that was used to create the security context.

861B6D0B Token header is not correct.

Explanation

The GSS-API token header is malformed or is corrupted.

User response

Verify that the token was not modified and then retry the request. Contact your service representative if the error persists.

861B6D0C Packet replayed in the wrong direction.

Explanation

The security mechanism specified in the token header is not correct.

User response

A GSS-API token was processed by the wrong partner. Tokens generated by the context initiator must be processed by the context acceptor. Tokens generated by the context acceptor must be processed by the context initiator.

861B6D51 Message is not within the current window.

Explanation

The message is not within the current receive window. This indicates that multiple messages are missing or an old message is being replayed. GSS-API is unable to determine whether the message has been processed previously.

User response

Verify that messages are being processed in the correct order and that no messages are being lost. Contact your service representative if the error persists.

861B6D52 Message is after the next message in the sequence.

Explanation

The message is within the current receive window but is later than the next message in the sequence. This indicates that one or more messages are missing.

User response

Verify that messages are being processed in the correct order and that no messages are being lost.

Contact your service representative if the error persists.

861B6D53 Message has already been received.

Explanation

The message is within the current receive window but has already been received. This indicates that an old message is being replayed.

User response

Verify that messages are being processed in the correct order and that no messages are being lost. Contact your service representative if the error persists.

861B6D54 Message is before the next message in the sequence.

Explanation

The message is within the current receive window but is earlier than the next message in the sequence. The message has not been received previously.

User response

Verify that messages are being processed in the correct order and that no messages are being lost. Contact your service representative if the error persists.

861B6D55 Token signature is not correct.

Explanation

The checksum computed using the token data does not match the checksum contained in the token.

User response

Verify that the token was not modified and then retry the request. Contact your service representative if the error persists.

861B6D56 Credential already contains a mechanism element.

Explanation

The **gss_add_cred_()** function was called to add a mechanism element to a credential. The credential already contains an element for the requested mechanism.

User response

Do not add an existing mechanism to a credential.

861B6D57 Context is expired.

Explanation

The GSS-API context has expired. This indicates that either the context lifetime has expired or the associated Kerberos ticket is no longer valid.

User response

Create a new GSS-API security context.

861B6D58 Token type is not correct.

Explanation

An attempt to decode a token failed due to incorrect API usage.

User response

If the token was created using the **gss_get_mic()** function, use the **gss_verify_mic()** function to process it. If the token was created using the **gss_wrap()** function, use the **gss_unwrp()** function to process it. If the token was created using the **gss_delete_sec_context()** function, use the **gss_process_context_token()** function to process it. If the token was created using the **gss_accept_sec_context()** function, use the **gss_init_sec_context()** function to process it.

861B6D59 Credential is expired.

Explanation

The GSS-API credential has expired. This indicates that either the credential lifetime has expired or the associated Kerberos ticket is no longer valid.

User response

Create a new GSS-API credential.

861B6D5A Required parameter is missing.

Explanation

A required parameter is not specified on a GSS-API function call.

User response

Specify the required parameter and then retry the request.

861B6D5B **Name is not a valid GSS-API name.**

Explanation

A *gss_name_t* parameter does not refer to a valid GSS-API name.

User response

Verify that the name parameter is correct. Names created by an application must be converted to the internal representation by calling the **gss_import_name()** function.

861B6D5C **No name is specified.**

Explanation

GSS_C_NO_NAME is specified on an API that requires that a name be provided.

User response

Specify a valid name and then retry the request.

861B6D5D **No mechanism is specified.**

Explanation

GSS_C_NO_OID is specified for the mechanism on an API that requires that a mechanism be specified.

User response

Specify a mechanism and then retry the request.

861B6D5E **Name type is not valid.**

Explanation

The name type specified on the **gss_import_name()** function call is not valid or is not supported by the current software level.

User response

Specify a valid name type and then retry the request.

861B6D5F **Name is not valid.**

Explanation

An attempt to convert a name to its internal representation was not successful, or, a different name was specified when **gss_init_sec_context()** was called to finish establishing the security context.

User response

Specify a valid name and then retry the request.

861B6D60 **Security mechanism is not valid.**

Explanation

The security mechanism is not valid or is not supported by the current software level.

User response

Specify a supported security mechanism.

861B6D61 **Status type is not valid.**

Explanation

The status type specified on the **gss_display_status()** function call is not valid.

User response

Specify a valid status type and then retry the request.

861B6D62 **Status value is not valid.**

Explanation

The status value specified on the **gss_display_status()** function call cannot be translated to an error message.

User response

Verify that the message catalog is installed and is available to the application.

861B6D63 **Object identifier encoding is not valid.**

Explanation

An object identifier does not have a valid encoding. Object identifiers are encoded using ASN.1 encoding rules. The string encoding consists of a series of blank-delimited or period-delimited numbers. The entire string is enclosed in braces.

User response

Specify a valid object identifier.

Kerberos database (numbers 95E73A00 - 95E73AFF)

95E73A01 Entry already exists in database.

Explanation

A request to add an entry to the Kerberos database failed because the entry already exists.

User response

Specify a name that does not already exist in the database.

95E73A02 Unable to store entry in database.

Explanation

An attempt to update the Kerberos database failed.

User response

Contact your service representative if the error persists.

95E73A03 Unable to read entry from database.

Explanation

An attempt to read an entry from the Kerberos database failed.

User response

Contact your service representative if the error persists.

95E73A04 Not authorized to perform requested operation.

Explanation

The requested operation cannot be performed because the client principal is not authorized.

User response

Ask your Kerberos administrator to grant the necessary authority to your Kerberos account.

95E73A05 Entry not found in database.

Explanation

The requested entry is not found in the Kerberos database.

User response

None

95E73A06 Incorrect use of wildcard character.

Explanation

A wildcard character is used incorrectly as part of a database search request.

User response

Specify a valid database search argument.

95E73A07 Database is in use by another process.

Explanation

The Kerberos database is locked by another process.

User response

Wait for the other process to release the Kerberos database and then retry the command.

95E73A08 Database modified during read operation.

Explanation

The Kerberos database was modified while a read request was being processed.

User response

Retry the failing request. Contact your service representative if the error persists.

95E73A09 Database record incomplete or corrupted.

Explanation

A record in the Kerberos database is incomplete or has been corrupted.

User response

Contact your service representative.

95E73A0A Recursive database lock request.

Explanation

The database support detected an attempt to lock the Kerberos database by a process that already holds the database lock.

User response

Contact your service representative.

95E73A0B Database is not locked.

Explanation

The database support detected an attempt to access the Kerberos database without holding the database lock.

User response

Contact your service representative.

95E73A0C Incorrect database lock mode.

Explanation

The database support detected an incorrect lock mode for a request to lock the Kerberos database.

User response

Contact your service representative.

95E73A0D Database is not initialized.

Explanation

An attempt to access the Kerberos database failed because the database has not been initialized.

User response

Contact your service representative.

95E73A0E Database is already initialized.

Explanation

An attempt to initialize the Kerberos database failed because the database has already been initialized.

User response

Contact your service representative.

95E73A0F Incorrect direction for key conversion.

Explanation

An incorrect direction flag was specified for an attempt to convert a Kerberos key.

User response

Contact your service representative.

95E73A10 No master key for database.

Explanation

The master key for the Kerberos database cannot be found.

User response

Contact your service representative.

95E73A11 Incorrect master key for database.

Explanation

The master key is not the correct database master key.

User response

Contact your service representative.

95E73A12 Key size is not valid.

Explanation

The key size for a database entry is not valid.

User response

Contact your service representative.

95E73A13 Unable to read stored master key.

Explanation

The database support is unable to read the stored master key for the Kerberos database.

User response

Contact your service representative.

95E73A14 Stored master key is corrupted.

Explanation

The stored master key for the Kerberos database has been corrupted.

User response

Contact your service representative.

95E73A15 Unable to lock database.

Explanation

The database support is unable to lock the Kerberos database.

User response

Contact your service representative.

95E73A16 Database corrupted.

Explanation

The Kerberos database is corrupted.

User response

Contact your service representative.

95E73A17 Unsupported database version.

Explanation

The Kerberos database version is not supported by the current software level.

User response

Upgrade to a software level that supports the database version.

95E73A18 Unsupported salt type.

Explanation

The password salt type is not supported by the current software level.

User response

Upgrade to a software level that supports the salt type.

95E73A19 Unsupported encryption type.

Explanation

The encryption type is not supported by the current software level.

User response

Upgrade to a software level that supports the encryption type.

95E73A1A Incorrect database creation flags.

Explanation

An attempt to create the Kerberos database failed because the creation request is not correct.

User response

Contact your service representative.

95E73AF0 Database record is too big.

Explanation

An attempt to create a Kerberos database entry failed because the record is too big. The Kerberos database has a maximum record size of 1024 bytes.

User response

Contact your service representative.

95E73AF1 Too many matching database records.

Explanation

A database search request resulted in more than 1000 matching records.

User response

Reduce the amount of data stored for the failing principal or policy.

95E73AF2 Duplicate database entry.

Explanation

A database entry cannot be created because an entry with the same name already exists.

User response

Use a different name for the new database entry.

95E73AF3 Database entry is still referenced.

Explanation

A database entry cannot be deleted because it is still refered to by other database entries.

User response

Modify the other database entries to remove the reference to the database entry that is to be deleted. Then retry the delete request.

95E73AF4 Database function is not supported.

Explanation

A database function was requested that is not supported by the current database.

User response

Do not request unsupported database functions.

95E73AF5 **Unknown security server.**

Explanation

The requested Kerberos security server is not defined.

User response

Either define the Kerberos security server or specify an existing security server.

95E73AF6 **Incompatible encryption type for the master key for the specified FIPS level.**

Explanation

This status code is returned to the caller when the NDBM master key is retrieved from the database or dump file for command processing and is not compatible with the FIPS level.

User response

None

Kerberos runtime codes (numbers 96C73A00 - 96C73CFF)

96C73A01 **Client entry in security registry has expired.**

Explanation

The client's entry in the registry database has expired.

User response

Restore the client's access and then retry the request.

96C73A02 **Server entry in security registry has expired.**

Explanation

The server's entry in the registry database has expired.

User response

Restore the server's access and then retry the request.

96C73A03 **Requested protocol version is not supported.**

Explanation

Kerberos request has been encoded using an unsupported protocol version.

User response

Ensure that the client and the server are at compatible software levels.

96C73A04 **Client key is encrypted using an old master key.**

Explanation

The client's key was encrypted using an old master key that is no longer contained in the registry database.

User response

Authenticate the client again in order to obtain a new ticket that is encrypted with the current master key.

96C73A05 **Server key is encrypted using an old master key.**

Explanation

The server's key was encrypted using an old master key that is no longer contained in the registry database.

User response

Obtain a new service ticket for the desired server.

96C73A06 **Client principal is not found in security registry.**

Explanation

The client principal in a Kerberos request was not found in the registry database.

User response

Add the client principal to the registry database and then retry the request.

96C73A07 **Server principal is not found in security registry.**

Explanation

The server principal in a Kerberos request was not found in the registry database. This error can occur if server principal is not defined in the security registry or if the Kerberos runtime is unable to obtain a TGT for the realm containing the server principal.

User response

Add the server principal to the registry database for the server realm. Ensure that a trust relationship exists between the client realm and the server realm. Then retry the request.

96C73A08 Server principal is not unique.

Explanation

The server principal in a Kerberos request has multiple entries in the registry database.

User response

This error should never be reported by the z/OS security server. Contact your service representative.

96C73A09 Key in security registry is not valid.

Explanation

A principal does not have an associated key in the registry database.

User response

Create a key for the principal and then retry the request.

96C73A0A Ticket is ineligible for postdating.

Explanation

A Kerberos request contains a "good-since" date that is in the future. The current security policy does not allowed postdated tickets.

User response

Retry the request without requesting a postdated ticket.

96C73A0B Effective ticket lifetime is too short.

Explanation

A Kerberos request specifies a lifetime that is less than the minimum lifetime allowed for a ticket.

User response

Retry the request specifying a longer lifetime.

96C73A0C Ticket request violates account administrative policy.

Explanation

A Kerberos request cannot be processed, for one of the following reasons:

- The client principal in the request is forbidden to have tickets by the administrative policy of the principal account.
- The server principal in the request is forbidden to be a server by the administrative policy of the principal account..
- The request asks for a ticket-granting ticket that allows postdating of tickets, which is not allowed by the client principal account.
- The request indicates that it is from a local client, but the contents of the request indicate that the request originates from a client in a foreign cell.
- The ticket-granting ticket in a request to the ticket-granting service has options that are not valid.

User response

Change the appropriate administrative policy to permit the request.

96C73A0D Ticket cannot be granted with requested properties.

Explanation

A Kerberos request cannot be processed because one or more of the ticket options are either not supported or are not allowed by the account policy. In the case of using the PKINIT authentication, this error occurs if the end date of the ticket exceeds that of the certificate set up for PKINIT.

User response

Change the account policy if desired and then retry the request. In the case of using the PKINIT authentication, retry the request with an earlier end date or use another certificate which has a later expiration date for PKINIT.

96C73A0E Encryption type is not supported.

Explanation

A Kerberos request contains an encryption type that is not supported or not compatible with the selected

FIPS level by the security server. The qualified encryption types for FIPS mode are:

- aes256-ctc-hmac-sha1-96
- aes128-cts-hmac-sha1-96
- des3-cbc-sha1
- aes128-cts-hmac-sha256-128
- aes256-cts-hmac-sha384-192

This return code will also happen when enctype des-hmac-sha1 is specified for FAST negotiation. Therefore, it is not just caused by FIPs level support from V2R4

User response

Ensure the encryption type used is a supported encryption type by each party (client, server, KDC) and if the party issuing the error has enabled FIPS, ensure the encryption type used is FIPS compliant.

96C73A0F Checksum type is not supported.

Explanation

A Kerberos request contains a checksum type that is not supported or compliant with the selected FIPS level by the security server. The qualified checksum types for FIPS mode are:

- hmac-sha1-96-aes256
- hmac-sha1-96-aes128
- hmac-sha1-des3
- hmac-sha256-128-aes128
- hmac-sha384-192-aes256

User response

Ensure the checksum type used is a supported checksum type by each party (client, server, KDC) and if the party issuing the error has enabled FIPS, ensure the checksum type used is FIPS compliant.

96C73A10 Preauthentication type is not supported.

Explanation

A Kerberos request contains a preauthentication type that is not supported by the security server. This error can occur if a request for a service ticket does not contain the ticket-granting ticket for the client.

User response

Provide the ticket-granting ticket when making a request to the ticket-granting service. Ensure that the client and server software levels are compatible.

96C73A11 Transited ticket not supported.

Explanation

A Kerberos request for a foreign principal contains a ticket-granting ticket granted by a foreign realm. The foreign realm is not a trust peer of the security server receiving the request.

User response

Use a ticket-granting ticket granted by the security server that will process the request or establish a trusted peer relationship between the two realms.

96C73A12 Client account is revoked.

Explanation

The "account valid" indicator is not set for the client account.

User response

Set the "account valid" indicator for the client account. Then retry the request.

96C73A13 Server account is revoked.

Explanation

The "account valid" indicator is not set for the server account.

User response

Set the "account valid" indicator for the server account. Then retry the request.

96C73A14 Ticket-granting ticket is expired.

Explanation

The expiration time for a ticket has been reached.

User response

Obtain a new ticket-granting ticket and then retry the request.

96C73A15 Client account is not yet valid.

Explanation

The "good-since" time for a client account is in the future.

User response

Wait until the client account becomes valid or change the "good-since" time for the account. Then retry the request.

96C73A16 Server account is not yet valid.

Explanation

The "good-since" time for a server account is in the future.

User response

Wait until the server account becomes valid or change the "good-since" time for the account. Then retry the request.

96C73A17 Password is expired.

Explanation

The principal's password has expired.

User response

Change the password for the principal and then retry the request.

96C73A18 Preauthentication failed.

Explanation

Preauthentication failed for a Kerberos request.

User response

Retry the failing request. Contact your service representative if the error persists.

96C73A19 Preauthentication required.

Explanation

An initial ticket request does not contain preauthentication data. The account policy for the principal requires preauthentication.

User response

Retry the request and provide preauthentication data.

96C73A1A Ticket is not for the requested server.

Explanation

The ticket supplied with a Kerberos request is not for the server specified in the request.

User response

Retry the request and provide the proper ticket.

96C73A1B Server requires ticket encrypted with session key.

Explanation

A request for a service ticket does not contain a session key. The account policy for the requested server requires that tickets be encrypted with a session key and not with the server key.

User response

Retry the request and provide the session key.

96C73A1C Transited path rejected.

Explanation

The KDC rejected the transited path encoded in the ticket-granting ticket provided with the request. This indicates that one of the realms involved in granting the ticket is not trusted by the KDC.

User response

Contact your administrator to update the trust relationships between the realms.

96C73A1D Service is not available.

Explanation

The requested service is not available.

User response

Retry the request. Contact your administrator if the problem persists.

96C73A1F Integrity check fails.

Explanation

The checksum computed using the decrypted message is not the same as the checksum contained within the message. This error can occur if the message has been modified or the wrong key was used to decrypt the message. The key can be incorrect if a password change has occurred for the principal. This error can also occur if the checksum contained

in the authenticator does not match the checksum computed using the supplied application data.

User response

Verify that the message was not modified. If an incorrect key was used, retry the request with the correct key. Contact your service representative if the error persists.

96C73A20 Ticket is expired.

Explanation

A Kerberos request cannot be completed because the associated ticket has expired.

User response

Obtain a new ticket and then retry the request.

96C73A21 Ticket is not yet valid.

Explanation

A Kerberos request cannot be completed because the start time for the associated ticket is in the future.

User response

Obtain a new ticket or wait until the current ticket is valid. Then retry the request.

96C73A22 Replay attempt detected.

Explanation

The Kerberos replay detection mechanism indicates that the received request is a replay of a prior request.

User response

Try the request again. Contact your service representative if the error persists.

96C73A23 Server name is not correct.

Explanation

A Kerberos request contains a ticket-granting ticket for another ticket-granting service.

User response

Obtain a ticket to the desired ticket-granting service and then retry the request. Contact your service representative if the error persists.

96C73A24 Client name is not correct.

Explanation

The client principal stored in the authenticator part of the Kerberos request does not match the client principal stored in the accompanying ticket.

User response

Ensure that the proper authenticator is used and then retry the request. Contact your service representative if the error persists.

96C73A25 Time differential exceeds maximum clock skew.

Explanation

The absolute difference between the timestamp in the message and the current system time is greater than the maximum clock skew value (normally 5 minutes). This problem can occur if the time on the client system is not the same as the time on the server system.

User response

Ensure that the time on both systems is synchronized properly.

96C73A26 Network address is not correct.

Explanation

The address of the message sender does not match any of the possible client addresses stored in the associated ticket.

User response

Obtain a new ticket containing the correct client addresses. Then retry the request. Contact your service representative if the error persists.

96C73A27 Message protocol version is not correct.

Explanation

The protocol version in a Kerberos message is not correct or is not supported by the security server.

User response

Ensure that the client and server software levels are compatible. Then retry the request. Contact your service representative if the error persists.

96C73A28 Message type is not correct.

Explanation

The message type in Kerberos message is not correct or is not supported by the security server.

User response

Ensure that the client and server software levels are compatible. Then retry the request. Contact your service representative if the error persists.

96C73A29 Message stream is modified.

Explanation

The messge packet sent to or received from the security server has been modified.

User response

Ensure that there are no communication problems between the client and the server. Then retry the request. Contact your service representative if the error persists.

96C73A2A Security message received in incorrect order.

Explanation

A security message was received that is not in the correct sequence.

User response

Ensure that there are no communication problems between the client and the server. Then retry the request. Contact your service representative if the error persists.

96C73A2B Illegal cross-realm ticket.

Explanation

An illegal cross-realm ticket was found when parsing an authorization service message.

User response

Obtain a ticket to the server that the request is attempting to access. Then retry the request. Contact your service representative if the error persists.

96C73A2C Service key version is not correct.

Explanation

The ticket associated with a Kerberos request specifies a server key type that is not correct or a server key version that is no longer contained in the registry database.

User response

Obtain a new ticket to the server that the request is attempting to access. Then retry the request. Contact your service representative if the error persists.

96C73A2D Service key is not available.

Explanation

The server key is not available for the server principal specified in a ticket. This error can occur if the server account was deleted after the ticket was granted.

User response

Recreate the server account and then retry the request.

96C73A2E Mutual authentication fails.

Explanation

A mutual authentication attempt failed.

User response

Retry the request. Contact your service representative if the error persists.

96C73A2F Message direction is incorrect.

Explanation

The message direction in a Kerberos message stream is incorrect. This error can occur if Kerberos peer services are being used and one of the messages is received out of order.

User response

Retry the request. Contact your service representative if the error persists.

96C73A30 Alternative authentication method required.

Explanation

A Kerberos request specifies an authentication method that is not supported by the security server. An alternative authentication method is required.

User response

Ensure that the client and server software levels are compatible. Then retry the request using an authentication method that is supported by the server.

96C73A31 Message sequence number is incorrect.

Explanation

A message was received containing an incorrect sequence number. This error can occur if one or more messages have been dropped by the communications network.

User response

Ensure that no communication errors are causing messages to be lost. Then retry the request. Contact your service representative if the error persists.

96C73A32 Checksum type is not appropriate.

Explanation

A checksum type has been selected that does not have the required properties for use with a security message. For example, generated checksums may not be unique using the selected checksum algorithm.

User response

Retry the request using an appropriate checksum type.

96C73A33 Transited path rejected.

Explanation

The application server rejected the transited path encoded in the service ticket provided with the request. This indicates that one of the realms involved in granting the ticket is not trusted by the server.

User response

Contact the application administrator to update the trust relationships between the realms.

96C73A34 Response too large for datagram.

Explanation

The response cannot be returned because it is too large for a UDP datagram.

User response

Change the application to use TCP instead of UDP for its communications.

96C73A3C Generic error occurs.

Explanation

An error has occurred that is not covered by any of the specific status codes defined for the Kerberos Key Distribution Center (KDC) component. Check the security server message log for more information about the cause of this error.

User response

Retry the request. Contact your service representative if the error persists.

96C73A3D Message field is too long.

Explanation

A Kerberos message contains a field that is longer than the maximum length supported by the security server.

User response

Ensure that the client and server software levels are compatible. Then retry the request. Contact your service representative if the error persists.

96C73A3E Client certificate is not acceptable.

Explanation

The client's certificate is not accepted for other unexpected reasons.

User response

Turn on the trace to find out more information.

96C73A3F KDC certificate is not acceptable.

Explanation

The KDC's certificate is not accepted for other unexpected reasons.

User response

Turn on the trace to find out more information.

96C73A40 Client certificate signature not valid.

Explanation

The client's certificate does not have a valid signature.

User response

Use a valid certificate.

96C73A41 **Client Diffie-Hellman key parameters not accepted.**

Explanation

The client uses the Diffie-Hellman key agreement method, but the key parameters provided does not satisfy the KDC's policy.

User response

Use the valid parameters.

96C73A46 **Client certificate could not be verified.**

Explanation

While validating the client's X.509 certificate, the KDC cannot build a certification path to validate the client's certificate.

User response

Use a valid certificate.

96C73A47 **Client certificate chain validation error occurred.**

Explanation

While processing the certification path, the KDC determines that the signature on one of the certificates in the signedAuthPack field is not valid.

User response

Make sure the signing chain of the certificate is correct.

96C73A48 **Client certificate chain contains a revoked certificate.**

Explanation

In the certification path validating the client's certificate, one or more of them are revoked.

User response

Make sure all the certificates in the chain are not revoked.

96C73A49 **Revocation status for the client certificate chain could not be determined.**

Explanation

The KDC attempts to determine the revocation status but is unable to do so.

User response

Make sure the revocation checking mechanism is provided for all the certificates in the signing chain.

96C73A4B **Kerberos client name does not match name bound to the client certificate.**

Explanation

The Kerberos client name in the AS-REQ request does not match a name bound by the KDC or if there is no binding found by the KDC.

User response

Use a certificate with the matching name in the Subject Alternate Name field.

96C73A4C **Kerberos KDC name does not match name bound to the KDC certificate.**

Explanation

The Subject Alternate Name in the KDC certificate used for PKINIT does not match the name bound to the KDC.

User response

Contact your KDC administrator to determine if the KDC certificate being used is correct, and what options are required in the client side configuration (krb5.conf). If the KDC certificate does not contain a Subject Alternative Name (SAN) OtherName of type id-pkinit-san, but instead contains a dNSname SAN, update the client side configuration file to include a pkinit_kdc_hostname entry in the appropriate realm section.

96C73A4D **Key purpose restricts client certificate usage.**

Explanation

The Extended Keyusage KeyPurposeId is required on the client certificate but it is not present, or the client certificate is restricted and not to be used for PKINIT authentication.

User response

Use a certificate with the required key purpose.

96C73A4E **Client certificate signature digest algorithm is not supported.**

Explanation

The digest algorithm used in the client certificate's signature is not acceptable by the KDC.

User response

Use a certificate with the signature generated by the supported digest algorithm.

96C73A4F **PKAuthenticator is missing the required paChecksum.**

Explanation

When the request is extended to negotiate hash algorithms, the client that does not want to use SHA1 sends the request in the extended message syntax without the paChecksum field. This error allows the client to retry with SHA1 if allowed by the local policy.

User response

None

96C73A50 **The signedData digest algorithm is not supported.**

Explanation

The digest algorithm used by the id-pkinit-authData is not acceptable by the KDC.

User response

Use the acceptable digest algorithm.

96C73A51 **The Public Key encryption delivery method is not supported.**

Explanation

The client does not want to use the Diffie-Hellman key delivery method but the KDC does not support the public key encryption key delivery method.

User response

Negotiate with the KDC.

96C73A52 **A well-known Kerberos principal name is used but not supported**

Explanation

A well-known Kerberos principal name was used in a protocol message when not supported by the receiving party.

User response

None

96C73A53 **A well-known Kerberos realm name is used but not supported**

Explanation

A well-known Kerberos realm name was used in a protocol message when not supported by the receiving party.

User response

None

96C73A54 **A reserved Kerberos principal name is used but not supported**

Explanation

A reserved Kerberos principal name was used in a protocol message, but is not supported by the receiving party.

User response

None

96C73A5D **An unknown critical FAST option was encountered**

Explanation

The KDC received an AS request that uses FAST pre-authentication data and the FAST pre-authentication data contains a critical FAST option that is not known to the KDC. By rule, the KDC must reject the request.

User response

Modify the request to remove the critical FAST option or use a different KDC that supports the critical FAST option.

96C73A81 **Lock request is not valid.**

Explanation

A request to lock a file does not specify a valid lock type.

User response

Specify a valid lock type and retry the request.

96C73A82 Unable to read password.

Explanation

An attempt to read a password from the terminal failed due to an input/output error. This error can also occur if no password is entered.

User response

Ensure that a valid input device is available to the application. Then retry the request. Contact your service representative if the error persists.

96C73A83 Password does not match expected value.

Explanation

The supplied password does not match the expected value. This error can occur if the same password is not specified for both the password prompt and the password validation prompt.

User response

Specify the same password for both the password prompt and the password validation prompt.

96C73A84 Password read is interrupted.

Explanation

An interrupt signal was received while reading the password from the terminal.

User response

Retry the request. Contact your service representative if the error persists.

96C73A85 Illegal character in component name.

Explanation

A component of a Kerberos name contains an illegal character.

User response

Ensure that a valid input device is available to the application. Then retry the request. Contact your service representative if the error persists.

96C73A86 Principal name is not valid.

Explanation

An error was detected while parsing the string representation of a principal name. The string representation consists of the name followed by an optional realm separated by "@" (the @ must be omitted if no realm is specified).

User response

Ensure that the principal name is properly formed.

96C73A87 Unable to open Kerberos configuration file.

Explanation

The Kerberos configuration file cannot be opened.

User response

Ensure that the configuration file exists and that the file permissions allow read access. Contact your service representative if the error persists.

96C73A88 Kerberos configuration file format is not valid.

Explanation

The Kerberos configuration file format is not valid. Refer to the administration guide for more information about the proper format of the file.

User response

Correct the file format errors and then retry the request.

96C73A89 Buffer is too small.

Explanation

The buffer specified on a Kerberos function call is too small to hold all of the return information.

User response

Specify a larger buffer and then retry the request. Contact your service representative if the error persists.

96C73A8A Message type is not valid.

Explanation

A Kerberos message cannot be encoded because the message type is not correct or is not supported by the current software level.

User response

Ensure that the software is at the correct level and then retry the request. Contact your service representative if the error persists.

96C73A8B Credentials cache name is not valid.

Explanation

The credentials cache name is not valid. A credentials cache name consists of an optional cache type followed by the name separated by a colon (:). The colon must be omitted if no cache type is specified.

User response

Specify a valid credentials cache name and then retry the request.

96C73A8C Credentials cache type is not valid.

Explanation

The credentials cache type is not valid or is not supported by the current software level. This error can occur if an application-specific cache type is specified and the cache type has not been registered with the Kerberos runtime.

User response

Specify a valid credentials cache type and then retry the request.

96C73A8D Matching credential is not found.

Explanation

A search of the credentials cache does not find a credential with the requested attributes.

User response

Change the search criteria and then retry the request.

96C73A8E End of credentials cache is reached.

Explanation

A read request has reached the end of the credentials cache.

User response

No action is required.

96C73A8F Required ticket is not supplied.

Explanation

A Kerberos request does not have a required ticket or the ticket is not complete.

User response

Refresh the credentials cache and then retry the request. Contact your service representative if the error persists.

96C73A90 Application server principal is not correct.

Explanation

The server principal in an application message does not match the principal name of the server receiving the request.

User response

Ensure that the application message is sent to the proper server. Contact your service representative if the error persists.

96C73A91 Application ticket is not valid.

Explanation

The ticket in an application message is not valid.

User response

Obtain a new service ticket and then retry the request. Contact your service representative if the error persists.

96C73A92 Principals do not match.

Explanation

An attempt to obtain credentials failed because the principal in the ticket-granting ticket does not match the target principal.

User response

Obtain a ticket-granting ticket for the correct principal and then retry the request.

96C73A93 Security server reply is modified.

Explanation

The values in the security server reply are not consistent with the request that was sent to the security server.

User response

Ensure that the security server reply was not modified. Contact your service representative if the error persists.

96C73A94 Clock skew in reply exceeds maximum value.

Explanation

The absolute difference between the timestamp in the security server reply and the local system time exceeds the maximum clock skew (normally 5 minutes).

User response

Ensure that the client and server system times are synchronized properly.

96C73A95 Realm mismatch in initial ticket request.

Explanation

The client realm is not the same as the principal realm in an initial ticket request.

User response

Ensure that the initial ticket request is sent to the security server for the client realm.

96C73A96 Encryption type is not valid.

Explanation

A Kerberos message specifies an encryption type that is not valid or is not supported by the current software level.

User response

Specify a valid encryption type and then retry the request. Contact your service representative if the error persists.

96C73A97 Key type is not valid.

Explanation

A Kerberos message specifies a key type that is not valid or is not supported by the current software level.

User response

Specify a valid key type and then retry the request. Contact your service representative if the error persists.

96C73A98 Encryption type is not correct.

Explanation

A Kerberos message is encrypted with the incorrect encryption type.

User response

Retry the request. Contact your service representative if the error persists.

96C73A99 Checksum type is not valid.

Explanation

A Kerberos message specifies a checksum type that is not valid or is not supported by the current software level.

User response

Specify a valid checksum type and then retry the request. Contact your service representative if the error persists.

96C73A9A Unable to locate security server.

Explanation

The Kerberos runtime is unable to locate the security server for the requested realm.

User response

Ensure that the requested realm is defined in the LDAP directory, the DNS name server, or the Kerberos configuration file. Ensure that the appropriate lookup mode is enabled in the Kerberos configuration file. Then retry the request.

96C73A9B Kerberos service is not defined.

Explanation

The requested Kerberos service is not defined or is not supported by the current software level.

User response

For an undefined service, define the desired service and then retry the request. For an unsupported service, upgrade to a software level that supports the service.

96C73A9C Unable to contact security server.

Explanation

The Kerberos runtime was unable to contact the security server for the requested realm.

User response

Ensure that the security server is running and is defined correctly in the LDAP directory, the DNS name server, or the Kerberos configuration file. Ensure that the appropriate lookup mode is enabled in the Kerberos configuration file. Then retry the request.

96C73A9D No local name found for principal.

Explanation

The Kerberos runtime was unable to translate a foreign principal to a local name.

User response

Ensure that a local name is defined for the foreign principal. Then try the request again.

96C73A9E Mutual authentication fails.

Explanation

A mutual authentication request was not successful.

User response

Retry the request. Contact your service representative if the error persists.

96C73A9F Replay cache type is already registered.

Explanation

The replay cache type is already registered with the Kerberos runtime.

User response

No action is required.

96C73AA0 Replay cache operation is unable to allocate memory.

Explanation

A memory allocation request failed for a replay cache control block.

User response

Increase the amount of memory available to the application and then retry the request.

96C73AA1 Replay cache type is not valid.

Explanation

The requested replay cache type is not valid or is not supported by the current software level. This error can occur if an application-specific cache type is requested and the cache type has not been registered with the Kerberos runtime.

User response

Register the cache type with the Kerberos runtime and then retry the request.

96C73AA2 Replay cache operation detects an unexpected error.

Explanation

An internal error was detected during a replay cache operation.

User response

Retry the request. Contact your service representative if the error persists.

96C73AA3 Message is a replay.

Explanation

The current message is a replay of a previous message.

User response

Ensure that your network has not been compromised and communication errors are not causing messages to be retransmitted. Contact your service representative if the error persists.

96C73AA4 Replay cache operation fails.

Explanation

A replay cache operation failed due to a file system error.

User response

Retry the request. Contact your service representative if the error persists.

96C73AA7 Replay cache file is truncated.

Explanation

An attempt to read from the replay cache file failed due to a premature end-of-file.

User response

Ensure that the replay cache file has not been modified. Contact your service representative if the error persists.

96C73AA8 Replay cache file operation is unable to allocate memory.

Explanation

A memory allocation request failed for a replay cache file request.

User response

Increase the amount of memory available to the application and then retry the request.

96C73AA9 Replay cache operation is unable to access file.

Explanation

An attempt to read from or write to the replay cache file failed.

User response

Ensure that the replay cache file is not damaged and that the file permissions allow the desired file access. Contact your service representative if the error persists.

96C73AAA Replay cache file system request fails.

Explanation

A file system request failed for the replay cache file.

User response

Ensure that the replay cache file is not damaged. Contact your service representative if the error persists.

96C73AAB Replay cache operation fails.

Explanation

A replay cache operation failed due to an internal error.

User response

Retry the request. Contact your service representative if the error persists.

96C73AAC Replay cache operation fails due to insufficient space.

Explanation

A replay cache operation failed due to insufficient space in the file system.

User response

Increase the available space in the file system and then retry the request.

96C73AB2 Cryptographic system detects an unexpected error.

Explanation

An unexpected error was detected by the Kerberos cryptographic system.

User response

Retry the request. Contact your service representative if the error persists.

96C73AB3 Key table name is not valid.

Explanation

The key table name is not valid. A key table name consists of an optional key table type followed by the name separated by a colon (:). The colon must be omitted if the key table type is not specified.

User response

Specify a valid key table name and then retry the request.

96C73AB4 Key table type is not valid.

Explanation

The key table type is not valid or is not supported by the current software level. This error can occur if an application-specific key table type is specified, and the key table type has not been registered with the Kerberos runtime.

User response

Register the key table type with the Kerberos runtime and then retry the request.

96C73AB5 Key table entry is not found.

Explanation

The requested key table entry was not found in the key table.

User response

List the entries in the key table file and if there is no entry for the principal used by the application then you will need to add one with the correct version number. If there is an entry already there, you will need to verify that the version number in the key table entry matches the version number for the same principal in the KDC database. If the KDC database has more than one entry for the principal, you need to match the entry with the highest version number.

96C73AB6 End of key table is reached.

Explanation

A key table read operation failed because the end of the key table was reached.

User response

No action is required.

96C73AB7 Key table does not support write operations.

Explanation

An attempt to add or delete a key table entry failed because the key table does not support write operations. This error can occur if the key table is opened using the FILE key table type instead of the WRFILE key table type.

User response

Change the key table type to WRFILE and then retry the request.

96C73AB8 Key table operation fails due to file system error.

Explanation

An attempt to read from or write to a key table failed due to a file system error.

User response

Ensure that the key table file was not damaged and then retry the request. Contact your service representative if the error persists.

96C73AB9 No ticket is found for ticket-granting service.

Explanation

An attempt to obtain a service or privilege ticket failed because no ticket was found for the ticket-granting service. This error can occur if the ticket-granting ticket (TGT) has expired and the application has not renewed it.

User response

Obtain a ticket-granting ticket for the desired ticket-granting service and then retry the request.

96C73ABA Key parity is not correct.

Explanation

A Data Encryption Standard (DES) key was supplied to the Kerberos cryptographic system. The DES key has incorrect parity. This error can occur if the DES key has been modified.

User response

Supply a valid DES key and then retry the request. Contact your service representative if the error persists.

96C73ABB Key does not provide adequate security.

Explanation

A Data Encryption Standard (DES) key was supplied to the Kerberos cryptographic system. The DES key is weak and does not provide adequate security for use by Kerberos.

User response

Supply a valid DES key and then retry the request. Contact your service representative if the error persists.

96C73ABC Encryption or checksum type is not supported.

Explanation

A cryptographic request failed because the requested encryption or checksum type is not supported.

User response

Specify a supported encryption or checksum type and then retry the request.

96C73ABD Key size is not valid.

Explanation

A key was supplied to the Kerberos cryptographic system. The key size is not valid.

User response

Specify a valid cryptographic key and then retry the request.

96C73ABE Encrypted message is too small.

Explanation

A request to decrypt a message failed because the message length is less than the minimum length for an encrypted message or the result buffer is smaller than the source buffer. A request to encrypt a message failed because the result buffer is too small. This error can also occur if the length of the initial vector is not correct for the requested encryption algorithm.

User response

For an error during decryption,ensure that the message has not been modified, and then retry the request. For an error during encryption, call the **krb5_c_encrypt_length()** routine to determine the required length for the result buffer. Contact your service representative if the error persists.

96C73ABF Credentials cache type is already registered.

Explanation

A request to register a credentials cache type failed because the credentials cache type is already registered with the Kerberos runtime.

User response

No action is required.

96C73AC0 Key table type is already registered.

Explanation

A request to register a key table type failed because the key table type is already registered with the Kerberos runtime.

User response

No action is required.

96C73AC1 Credentials cache file operation fails.

Explanation

A file system request failed for the credentials cache file.

User response

Ensure that the credentials cache file was not damaged and then retry the request. Contact your service representative if the error persists.

96C73AC2 Credentials cache file cannot be accessed.

Explanation

The credentials cache file cannot be accessed due to a file system error.

User response

Ensure that the credentials cache file exists and the file permissions permit the application to access the file. Then retry the request. Contact your service representative if the error persists.

96C73AC3 Credentials cache file does not exist.

Explanation

The credentials cache file does not exist.

User response

Specify an existing credentials cache file and then retry the request.

96C73AC4 Credentials cache operation detects an unexpected error.

Explanation

A credentials cache operations failed due to an internal error.

User response

Retry the request. Contact your service representative if the error persists.

96C73AC5 Credentials cache write operation fails.

Explanation

A credentials cache write request failed due to a file system error.

User response

Retry the request. Contact your service representative if the error persists.

96C73AC6 **Credentials cache operation is unable to allocate memory.**

Explanation

A credentials cache request was unable to allocate memory.

User response

Increase the amount of memory available to the application and then retry the request. Contact your service representative if the error persists.

96C73AC7 **Credentials cache format is not valid.**

Explanation

Credentials cache format is not valid.

User response

Specify a credentials cache format that is supported and then retry the request. Contact your service representative if the error persists.

96C73AC8 **Credentials request specifies incorrect options.**

Explanation

A security credentials request specifies options that are not correct or are not supported by the current software level.

User response

Specify valid options and then retry the request. Contact your service representative if the error persists.

96C73AC9 **Credentials request does not contain second ticket.**

Explanation

A credentials request cannot be processed because two tickets are required, and the request does not contain the second ticket.

User response

Provide the second ticket and then retry the request. Contact your service representative if the error persists.

96C73ACA **No credentials are available.**

Explanation

A Kerberos function was called, but no credentials are provided by the caller.

User response

Provide the required credentials and then retry the request. Contact your service representative if the error persists.

96C73ACB **Incorrect authentication protocol version.**

Explanation

The authentication protocol version in the message stream is not supported by the **krb5_recvauth()** routine.

User response

Ensure that compatible levels of the Kerberos runtime are installed on the local and remote systems. Contact your service representative if the error persists.

96C73ACC **Incorrect application version identifier.**

Explanation

The application version identifier specified for the **krb5_recvauth()** routine does not match the application version identifier specified for the **krb5_sendauth()** routine.

User response

Specify the same application version identifier string.

96C73ACD **Unrecognized response received from remote application**

Explanation

The remote application returned an unrecognized response.

User response

Ensure that compatible levels of the Kerberos runtime are installed on the local and remote systems. Contact your service representative if the error persists.

96C73ACE **Authentication rejected by application server.**

Explanation

The remote application server has rejected the client authentication.

User response

Contact the application support programmer.

96C73ACF **Preauthentication type is not valid.**

Explanation

The preauthentication type is not valid or is not supported by the current software level.

User response

Specify a valid preauthentication type and then retry the request. Contact your service representative if the error persists.

96C73AD0 **No preauthentication key is provided.**

Explanation

The Kerberos runtime is unable to encrypt the preauthentication data because the encryption key is not provided by the application.

User response

Provide the required encryption key and then retry the request. Contact your service representative if the error presists.

96C73AD1 **Preauthentication fails.**

Explanation

A preauthentication request failed. This error can occur if the incorrect principal key is specified. It can also occur if the system clocks on the client and server systems are not within 5 minutes of each other.

User response

Ensure that the system clocks are synchronized and the correct principal key is specified. Then retry the request. Contact your service representative if the error persists.

96C73AD2 **Replay cache version number is not valid.**

Explanation

The replay cache version number is not valid or is not supported by the current software level.

User response

Ensure that the replay cache file has not been modified and then retry the request. Contact your service representative if the error persists.

96C73AD3 **Credentials cache version number is not valid.**

Explanation

The credentials cache version number is not valid or is not supported by the current software level.

User response

Ensure that the credentials cache file has not been modified and then retry the request. Contact your service representative if the error persists.

96C73AD4 **Key table version number is not valid.**

Explanation

The key table version number is not valid or is not supported by the current software level.

User response

Ensure that the key table file has not been modified and then retry the request. Contact your service representative if the error persists.

96C73AD5 **Address type is not valid.**

Explanation

The network address type is not valid or is not supported by the current software level.

User response

Provide a valid network address type and then retry the request. Contact your service representative if the error persists.

96C73AD6 **Replay detection requires a replay cache.**

Explanation

Replay detection was requested but no replay cache is available.

User response

Set up a replay cache and then retry the request. Contact your service representative if the error persists.

96C73AD7 Host name is not defined.

Explanation

An attempt to obtain the network host entry using the **gethostbyname()** or **gethostbyaddr()** function failed. The most likely cause of this error is that the host name or network address is not defined to the domain name service.

User response

Ensure that the host name is defined to the domain name service and that name resolution is working. Then retry the request. Contact your service representative if the error persists.

96C73AD8 Host realm is not defined.

Explanation

The realm corresponding to a host name cannot be determined.

User response

Ensure the host realm is defined in the Kerberos configuration file and then retry the request. Contact your service representative if the error persists.

96C73AD9 Name cannot be converted to service principal.

Explanation

A name was provided that cannot be converted to a service principal because no conversion exists for the name type.

User response

Provide a name that can be converted to a service principal and then retry the request. Contact your service representative if the error persists.

96C73ADA Initial ticket response generated by Kerberos Version 4.

Explanation

The response to an initial ticket response is in Kerberos Version 4 format. The Kerberos runtime supports only Kerberos Version 5 formats.

User response

Send the initial ticket request to a Kerberos Version 5 security server. Contact your service representative if the error persists.

96C73ADB Security server is not defined for requested realm.

Explanation

The Kerberos runtime is unable to locate the security server for the requested realm. This error can occur if the requested realm is not defined or if the security server host name cannot be resolved to a network address.

User response

Ensure the security server is defined in either the LDAP directory or the Kerberos configuration file and then retry the request. Contact your service representative if the error persists.

96C73ADC Ticket-granting ticket does not allow ticket forwarding.

Explanation

An attempt to obtain a forwarded ticket failed because the ticket-granting ticket (TGT) provided with the request does not allow ticket forwarding.

User response

Provide a ticket that allows ticket forwarding and then retry the request. Contact your service representative if the error persists.

96C73ADD Principal name is not correct for forwarding credentials.

Explanation

The principal name is not correct for forwarding credentials. The principal name type must be KRB5_NT_SRV_HST. When creating forwarded credentials for use with GSS-API delegation, the target name must have been imported by specifying GSS_C_NT_HOSTBASED_SERVICE as the name type on the **gss_import_name()** function call.

User response

Create the principal name in the proper format and then retry the request. Contact your service representative if the error persists.

96C73ADE Request loop is detected while obtaining initial ticket.

Explanation

The **krb5_get_in_tkt()** function detects a request loop while obtaining the intermediate ticket-granting tickets necessary to process the request. This error can occur if the peer trust relationships are not correct between the security servers in the intermediate realms.

User response

Retry the request. Contact your service representative if the error persists.

96C73ADF **No default realm is specified in the configuration file.**

Explanation

The Kerberos configuration file does not define a default realm.

User response

Define a default realm in the configuration file. Then retry the request.

96C73AE1 **Key table name is too long.**

Explanation

The key table name is too long.

User response

Specify a valid key table name and then retry the request.

96C73C00 **Unable to load code page table.**

Explanation

The Kerberos runtime is unable to load the tables that are used to convert text strings between the network code page and the local code page.

User response

Contact your service representative if the error persists.

96C73C01 **Unable to convert text string.**

Explanation

The Kerberos runtime is unable to convert a text string to/from the network code page.

User response

Contact your service representative if the error persists.

96C73C02 **Unable to allocate memory.**

Explanation

The Kerberos runtime was unable to allocate memory for a control block.

User response

Increment the memory available to the application and then retry the request. Contact your service representative if the error persists.

96C73C03 **Unable to obtain the current time.**

Explanation

The Kerberos runtime is unable to obtain the current time of day.

User response

Verify that the system time provider is running and is configured properly. Contact your service representative if the error persists.

96C73C04 **Key table file specification is not valid.**

Explanation

The key table file specification is not valid. Either the file name is not correct or the file cannot be accessed.

User response

Verify that the file name is correct and that the file permissions allow access by the application. Then retry the request. Contact your service representative if the error persists.

96C73C05 **Key table operation encounters unexpected error.**

Explanation

An unexpected error was detected during a key table operation.

User response

Contact your service representative if the error persists.

96C73C06 **Unable to create socket.**

Explanation

The Kerberos runtime was unable to create a socket because the **socket()** function failed.

User response

Contact your service representative if the error persists.

96C73C07 **Unable to obtain local address information.**

Explanation

The Kerberos runtime was unable to obtain local address information for a socket because the **ioctl()** function failed.

User response

Contact your service representative if the error persists.

96C73C08 **Control block validation fails.**

Explanation

A Kerberos control block does not contain the proper identifier value.

User response

Verify that the control block was not modified. Contact your service representative if the error persists.

96C73C09 **Invalid parameter specified on function call.**

Explanation

A parameter specified on a Kerberos function call is not correct.

User response

Verify that the proper parameters are specified and then retry the request. Contact your service representative if the error persists.

96C73C0A **Unsupported Kerberos function requested.**

Explanation

A Kerberos function was requested that is not implemented in the current software configuration.

User response

Change the application to request a supported function.

96C73C0B **Replay cache file does not exist.**

Explanation

The replay cache file does not exist.

User response

Verify that the replay cache file exists and that the file permissions allow access by the application.

96C73C0C **Not authorized to access credentials cache.**

Explanation

The application is not authorized to access the credentials cache.

User response

Verify that the credentials cache access permissions allow access by the application.

96C73C0D **Not authorized to access replay cache.**

Explanation

The application is not authorized to access the replay cache.

User response

Verify that the file permissions allow access by the application.

96C73C0E **Not authorized to access key table.**

Explanation

The application is not authorized to access the key table.

User response

Verify that the file permissions allow access by the application.

96C73C0F **Data privacy service is not available.**

Explanation

The requested cryptographic algorithm is not available. This error can occur if the software algorithm is not installed or if a cryptographic algorithm requires the use of a hardware cryptographic processor that is not available on the current system.

User response

Ensure that the proper hardware and software is installed for the cryptographic algorithm.

96C73C10 Unable to retrieve message *msg-identifier* from the message catalog.

Explanation

The security runtime is unable to retrieve message text from the message catalog.

User response

Ensure the NLSPATH environment variable is set properly and then retry the request. Contact your service representative if the error persists.

96C73C11 Unable to contact server.

Explanation

The Kerberos runtime is unable to contact the server providing the requested service.

User response

Ensure that the server is running and is defined correctly in the LDAP directory, the DNS name server, or the Kerberos configuration file. Ensure that the appropriate lookup mode is enabled in the Kerberos configuration file. Then retry the request.

96C73C12 Key version value is not supported by the key table format.

Explanation

The key version value is not within the range supported by the key table format. The current key table implementations store the key version as a 1-byte value. This means the key version must be between 1 and 255 when stored in a key table.

User response

Use a key version between 1 and 255. If the Kerberos database does not allow the key version to be reset for a principal, you must delete the principal from the database and then add it again in order to reset the key version to 1.

96C73C13 Unable to send data to remote application.

Explanation

The Kerberos runtime is unable to send data to the remote application.

User response

Ensure that there are no network problems and that the remote application is running. Contact your service representative if the error persists.

96C73C14 Unable to receive data from remote application.

Explanation

The Kerberos runtime is unable to receive data from the remote application.

User response

Ensure there are no network problems and the remote application is running. Contact your service representative if the error persists.

96C73C15 Connection closed by remote application.

Explanation

The Kerberos runtime is unable to receive data from the remote application because the connection has been closed. This error can occur if the local system is not authorized to establish a connection with the remote system.

User response

Ensure that there are no network problems, that the remote application is running, and that the local system is authorized to establish the connection. Contact your service representative if the error persists.

96C73C16 Password is too long.

Explanation

The password is longer than 128 characters.

User response

Enter a shorter password.

96C73C17 Response too large.

Explanation

The response cannot be returned because it is too large for the buffer provided.

User response

Contact your service representative.

96C73C1B PKINIT context is not initialized.

Explanation

PKINIT configuration is not valid.

User response

Correct the configuration.

96C73C1C Unable to open key ring, key token, or key database.

Explanation

Unable to open key ring, key token, or key database for PKINIT.

User response

Make sure that the key ring, key token, or key database exists.

96C73C1D Error reading certificate record.

Explanation

Certificates can not be retrieved from the key ring, key token, or key database.

User response

Make sure that the key ring, key token, or key database is set up properly with certificates.

96C73C1E Key ring, key token, or key database configuration error.

Explanation

The key ring, key token, or key database set up for PKINIT is not valid. PKINIT is not used for preauthentication.

User response

Ensure that the key ring, key token, or key database is specified with the correct value.

96C73C1F No certificate records found.

Explanation

The key ring, key token, or key database set up for PKINIT does not contain any certificates. PKINIT is not used for preauthentication.

User response

Ensure that the key ring, key token, or key database contains the valid certificates.

96C73C20 The default certificate does not have an associated private key.

Explanation

The default certificate set up for PKINIT does not have an associated private key in key ring, key token, or key database. PKINIT is not used for preauthentication.

User response

Ensure that default certificate has the associated private key in the key ring, key token, or key database.

96C73C21 The default certificate is self signed.

Explanation

The default certificate set up for PKINIT is a self-signed certificate. PKINIT is not used for preauthentication.

User response

Ensure that default certificate is signed by a trusted CA.

96C73C22 The default certificate is not an RSA key certificate.

Explanation

The key type of the default certificate set up for PKINIT is not RSA. PKINIT is not used for preauthentication.

User response

Ensure that default certificate is an RSA certificate.

96C73C23 The default certificate is expired.

Explanation

The default certificate set up for PKINIT is expired. PKINIT is not used for preauthentication.

User response

Ensure that default certificate is not expired.

96C73C24 Issuer certificate is not a CA.

Explanation

A certificate in the issuer chain is not a CA certificate. PKINIT is not used for preauthentication.

User response

Ensure that the chain contains CA certificates.

96C73C25 No signing certificates found.

Explanation

The key ring, key token, or key database set up for PKINIT does not contain any certificates that can be used for PKINIT. PKINIT is not used for preauthentication.

User response

Ensure that the key ring, key token, or key database contains at least one certificate that can be used for PKINIT.

96C73C26 No root CA certificates found.

Explanation

The key ring, key token, or key database set up for PKINIT does not contain the root certificate of the other party's certificate for validation. PKINIT is not used for preauthentication.

User response

Ensure that the key ring, key token, or key database contains the root certificate of the other party's certificate.

96C73C27 Error copying a certificate or private key.

Explanation

The certificate or the associated private key cannot be retrieved from the key ring, key token, or key database.

User response

Ensure that the process has the authority to access the certificate or key.

96C73C28 Error determining if a certificate is a CA certificate.

Explanation

Error occurs when the process is trying to determine if a certificate in the chain for PKINIT is a CA certificate.

User response

None

96C73C29 Diffie-Hellman minimum bits configuration error.

Explanation

PKINIT configuration for the Diffie-Hellman key size is not valid. Diffie-Hellman key size 2048 bits is used for processing

User response

Correct the value for the Diffie-Hellman key size for future use.

96C73C2A Revocation checking configuration error.

Explanation

PKINIT configuration for certificate revocation checking mechanism is not valid. No revocation check is performed.

User response

Correct the value for the certificate revocation checking mechanism.

96C73C2B LDAP server configuration error.

Explanation

PKINIT configuration for certificate revocation checking mechanism using LDAP is not valid. No revocation check is performed.

User response

Correct the value for the certificate revocation checking mechanism.

96C73C2C Missing LDAP server configuration.

Explanation

PKINIT configuration for certificate revocation checking mechanism is trying to use LDAP. However the LDAP server is not specified. No revocation check is performed.

User response

Specify the LDAP server value if LDAP is the revocation checking mechanism.

96C73C2D **RSA protocol configuration error.**

Explanation

PKINIT configuration for the RSA protocol is not valid. The Diffie-Hellman protocol is used for processing.

User response

Correct the value for the RSA protocol for future use.

96C73C2E **The default certificate signature algorithm does not meet the FIPS requirement.**

Explanation

The signature algorithm of the default certificate in the client key ring set up for PKINIT does not meet FIPS requirement.

User response

Ensure that the signature algorithm of the default certificate in the client key ring is FIPS compliant if running in FIPS mode. Rerun the application once the correction is made. For a description of the appropriate certificate signature algorithm for a given FIPS level, see "Configuration of encryption types and FIPS level" on page 17.

96C73C2F **The default certificate key size does not meet the FIPS requirement.**

Explanation

The default certificate key size set up in the client key ring for PKINIT does not meet FIPS requirement.

User response

Ensure that the default certificate key size in the key ring is FIPS compliant if running in FIPS Mode. Rerun the application once the correction is made. For a description of the appropriate key size for a given FIPS level, see "Configuration of encryption types and FIPS level" on page 17.

96C73C30 **Incompatible encryption type specified for the requested FIPS level.**

Explanation

None of the encryption types specified by default_tgs_enctypes or default_tkt_enctypes is compliant with the level specified by fipslevel in the Kerberos configuration file. The qualified encryption types for FIPS mode are:

- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha1-96
- des3-cbc-sha1
- aes128-cts-hmac-sha256-128
- aes256-cts-hmac-sha384-192

User response

Change the encryption type or the FIPS level accordingly. Rerun the application once the correction is made.

96C73C31 **Incompatible checksum type specified for the requested FIPS level.**

Explanation

An application or command attempted to use a checksum type that does not meet the FIPS level established for the environment. CRC32, MD4, and MD5 checksum types are not allowed in a FIPS environment.

User response

Determine if the application or command can be configured to use an allowable checksum type in a FIPS environment or disable FIPS for the application or command.

96C73C32 **The certificate key size does not meet the FIPS requirement.**

Explanation

A certificate in the KDC key ring contains at least one certificate that has a public key size that does not meet the FIPS level that the KDC is running.

User response

Use a certificate that has the appropriate key size. For a description of the appropriate public key size for a given FIPS level, see "Configuration of encryption types and FIPS level" on page 17.

96C73C33 **The certificate signature algorithm does not meet the FIPS requirement.**

Explanation

A certificate in the KDC key ring contains at least one certificate that has a signature algorithm that does not meet the FIPS level that the KDC is running.

User response

Use a certificate that has the appropriate signature algorithm. For a description of the appropriate certificate signature algorithm for a given FIPS level, see "Configuration of encryption types and FIPS level" on page 17.

96C73C34 The certificate chain does not meet the FIPS requirement.

Explanation

During a PKINIT authentication request, the KDC encountered a certificate that does not comply with the FIPS level it is running.

User response

Determine which of the client and its intermediate CA signing certificates are not compliant with the KDC FIPS level and obtain new certificates that meet the KDC FIPS level. See "Configuration of encryption types and FIPS level" on page 17 for the public key size and signature algorithm requirements for a given FIPS level.

96C73C35 Encryption type in KDC response is incompatible with client FIPS level.

Explanation

The KDC uses an incompatible encryption type to build the response for the client when the client is running in FIPS mode.

User response

Contact the system administrator to ensure the KDC uses an encryption type that is compliant with the client's FIPS level, or disable FIPS in the client Kerberos configuration.

96C73C36 The KDC AS-REP message is missing KRB-PKINIT-KX pre-authentication data

Explanation

The kinit command or krb5_get_in_tkt_with_pkinit API received an AS-REP message from the KDC for an anonymous PKINIT request that did not contain the required KRB-PKINIT-KX pre-authentication data.

User response

Contact the KDC administrator to troubleshoot the issue or use a different KDC that supports anonymous PKINIT.

96C73C37 Ticket rejected for FAST negotiation protocol violation

Explanation

The kinit command or krb5_get_in_tkt_with_password API received an AS-REP message from the KDC that did not contain FAST negotiation pre-authentication data in the encrypted part of the response. This is a violation of the FAST negotiation protocol.

User response

Contact the KDC administrator to troubleshoot the issue.

Profile operations codes (numbers AACA6000 - AACA60FF)

AACA6002 Profile section is not found.

Explanation

A profile read request failed because the requested profile section was not found.

User response

No action is required.

AACA6003 Profile relation is not found.

Explanation

A profile read request failed because the requested profile relation was not found.

User response

No action is required.

AACA6004 Profile node is not a section node.

Explanation

A profile add was requested, but the supplied node is not a section node.

User response

Provide a section node and retry the request.

AACA6005 Profile section node has a value.

Explanation

A profile section node is not allowed to have a value.

User response

Verify that the configuration file is formatted properly. Contact your service representative if the error persists.

AACA6009 Profile section does not have a parent.

Explanation

The parent pointer in a profile section node is not valid.

User response

Verify that the profile entries in memory are not corrupted. Contact your service representative if the error persists.

AACA600A Profile section is not correct.

Explanation

A profile section is not formatted correctly.

User response

Verify that the configuration file is formatted properly. Contact your service representative if the error persists.

AACA600B Profile relation is not correct.

Explanation

A profile relation is not formatted correctly.

User response

Verify that the configuration file is formatted properly. Contact your service representative if the error persists.

AACA600C Extra closing brace is specified.

Explanation

An extra closing brace was found in a list of relations while processing a profile definition.

User response

Verify that the configuration file is formatted properly. Contact your service representative if the error persists.

AACA600D Opening brace is missing.

Explanation

An opening brace is missing for a list of relations while processing a profile definition.

User response

Verify that the configuration file is formatted properly. Contact your service representative if the error persists.

AACA6013 Profile name set is not correct.

Explanation

The set of names specified for a profile lookup operation is not correct. The name set must include at least the section name and the relation name.

User response

Specify a valid name set and retry the request.

AACA6014 No profile is available.

Explanation

No Kerberos profile was found.

User response

Create a Kerberos configuration file if needed.

Chapter 7. Messages

This chapter contains three sets of messages:

- Messages from the Kerberos runtime (EUVF02000 through EUVF03999)
- Messages from the security server (EUVF04000 through EUVF05999)
- Messages from Kerberos commands (EUVF06000 through EUVF06999).

Messages are listed in numerical order.

Kerberos runtime messages (numbers EUVF02000 - EUVF03999)

EUVF02001E Unable to read a required input parameter.

Explanation

An attempt to read a required input parameter failed.

User response

Verify that all required parameters are specified and then retry the request. Contact your service representative if the error persists.

EUVF02002E Unable to modify a required output parameter.

Explanation

An attempt to modify a required output parameter fails.

User response

Verify that all required parameters are specified and then retry the request. Contact your service representative if the error persists.

EUVF02003E Parameter is incorrectly structured.

Explanation

A parameter is incorrectly structured.

User response

Verify that all parameters are correct and then retry the request. Contact your service representative if the error persists.

EUVF02004E Security mechanism is not supported.

Explanation

The security mechanism is not valid or is not supported by the current software level.

User response

Specify a security mechanism that is supported by the current software level. Contact your service representative if the error persists.

EUVF02005E Name value is not valid.

Explanation

The specified name value is not valid.

User response

Specify a valid name value and then retry the request. Contact your service representative if the error persists.

EUVF02006E Name type is not valid.

Explanation

The specified name type is not valid.

User response

Specify a valid name type and then retry the request. Contact your service representative if the error persists.

EUVF02007E Channel bindings do not match token bindings.

Explanation

The channel bindings specified on the function call do not match the channel bindings contained in the input token.

User response

Ensure that the security context initiator is specifying the correct channel bindings. Contact your service representative if the error persists.

EUVF02008E Status value is not valid.

Explanation

The status value passed to the **gss_display_status()** function is not a valid GSSAPI status.

User response

Verify that the status value is not modified. Contact your service representative if the error persists.

EUVF02009E Token signature is not correct.

Explanation

The checksum computed from the token data does not match the signature contained in the token.

User response

Verify that the token is not modified. Contact your service representative if the error persists.

EUVF02010E No security credential supplied.

Explanation

Either the supplied security credential is not valid for context acceptance or the credential handle is not valid.

User response

Provide a valid security credential that can be used to accept a security context.

EUVF02011E No security context established.

Explanation

The requested function requires a security context but no security context is supplied.

User response

Provide a security context and then retry the request.

EUVF02012E Token is not valid.

Explanation

The token contents are not valid.

User response

Verify that the token is not modified and then retry the request. Contact your service representative if the error persists.

EUVF02013E Security credential is not valid.

Explanation

The security credential is not valid because internal consistency checks fail.

User response

Verify that no storage overlay has occurred and then retry the request. Contact your service representative if the error persists.

EUVF02014E Security credential is expired.

Explanation

Either the security credential lifetime is expired or the associated ticket is no longer valid.

User response

Create a new security credential and then retry the request. Contact your service representative if the error persists.

EUVF02015E Security context is expired.

Explanation

Either the security context lifetime is expired or the associated security credential is no longer valid.

User response

Create a new security context and then retry the request. Contact your service representative if the error persists.

EUVF02016E Security mechanism detects error.

Explanation

The security mechanism detects an error. The minor status code provides additional information concerning the error. The minor status code is zero if the error cannot be isolated to a single security mechanism.

User response

Refer to the minor status code to determine the action to be taken. Contact your service representative if the error persists.

EUVF02017E Quality of protection value is not valid.

Explanation

The quality of protection value is not valid or is not supported by the current software level.

User response

Specify a valid quality of protection value and then retry the request. Contact your service representative if the error persists.

EUVF02018E Requested operation is not authorized.

Explanation

The requested operation is not authorized by the associated security credential.

User response

Obtain the necessary authorization and then retry the request. Contact your service representative if the error persists.

EUVF02019E Requested operation is not available.

Explanation

The requested operation is not provided by the current software level.

User response

Upgrade the software to a level that supports the requested operation.

EUVF02020E Duplicate credential element requested.

Explanation

The requested credential element is already present in the security credential.

User response

None required.

EUVF02021E Name contains multiple mechanism elements.

Explanation

The supplied name contains elements for multiple mechanisms. The requested operation requires a name with a single mechanism element.

User response

Provide a valid name and then retry the request.

EUVF02022I Response token required from peer application.

Explanation

To complete the security context, the current function must be called again with the response token obtained from the peer application.

User response

None required.

EUVF02023W Message is a duplicate of one already received.

Explanation

The message is valid and is a duplicate of one that was already received.

User response

Application specific.

EUVF02024W A more recent message was already received.

Explanation

The current message is old. A more recent message has already been received. The message validity period has expired, so the routine cannot determine whether the message is a duplicate of one that was already received.

User response

Application specific.

EUVF02025W Message received out of sequence.

Explanation

The message is valid but an earlier message in the sequence has not been received.

User response

Application specific.

EUVF02026W Skipped predecessor message detected.

Explanation

The message is valid but a later message in the sequence has already been received.

User response

Application specific.

EUVF02027E Kerberos control block validation fails: Expected *exp-value*, Actual *act-value*.

Explanation

Kerberos internal control blocks have a unique identifier for each type of control block. The Kerberos runtime detects a mismatch between the expected identifier and the actual identifier. This error can occur if storage has been overlaid or modified.

User response

Contact your service representative.

EUVF02028E The *name* system function detects an error. *error-text*.

Explanation

A system function detects an error. Refer to the documentation for the failing system function to obtain more information about the cause of the failure.

User response

Contact your service representative if the error persists.

EUVF02029E The *name* system function detects an error on *filename*. *error-text*.

Explanation

A system function detected an error while processing the indicated file.

User response

Contact your service representative if the error persists.

EUVF02030E Syntax error on line *number* of *filename*. Error *number*: *error-text*.

Explanation

A syntax error was detected while processing the Kerberos profile.

User response

Correct the line in error and restart the application.

EUVF02031E The *name* cryptographic function detects an error: Return code *rtn-code*, Reason code *rsn_code*.

Explanation

A cryptographic function detected an error. Refer to *ICSF Application Programmer's Guide* for a description of the failing function and an explanation of the return code and reason code.

User response

Contact your service representative if the error persists.

EUVF02032E The *name* network function detects an error, *error-text*.

Explanation

A network function detected an error. Refer to the documentation for the failing function to obtain more information about the cause of the failure.

User response

Contact your service representative if the error persists.

EUVF02033R Enter password:

Explanation

The security runtime needs the user password to complete a request.

User response

Enter your password.

EUVF02034E Unable to initialize Kerberos GSS-API mechanism: Error *code*.

Explanation

An error occurred during the initialization of the Kerberos GSS-API mechanism.

User response

Contact your service representative if the error persists.

EUVF02035E Mutex operation fails. *error-text*.

Explanation

A mutex operation failed. error-text is the message text associated with the error code.

User response

Contact your service representative if the error persists.

EUVF02036I Call stack traceback called from *filename* at line *linenumber*.

Explanation

A call stack trace was requested by the security runtime. This message identifies the source module name and source line number that made the request.

User response

No action is required.

EUVF02037I Called from *function* at offset *displacement*.

Explanation

A call stack trace was requested by the security runtime. This message is issued for each entry in the call stack and identifies the offset within the function that issued the call.

User response

No action is required.

EUVF02038I IBM Kerberos dump created.

Explanation

A dump has been created by the security runtime. The dump was created in the directory specified by the _CEEDUMP_DIR environment variable.

User response

Contact your service representative and provide the dump file.

EUVF02039E Incorrect request format.

Explanation

The password change protocol packet is not formatted correctly. The result string returned by the password server contains additional information about the error.

User response

Contact your service representative if the error persists.

EUVF02040E Password server error.

Explanation

The password server detected an error. The result string returned by the password server contains additional information about the error.

User response

Contact your service representative if the error persists.

EUVF02041E Authentication error.

Explanation

The authentication information supplied with a password change request is not correct. The result string returned by the password server contains additional information about the error.

User response

Contact your service representative if the error persists.

EUVF02042E Password change rejected.

Explanation

The password change request is rejected by the password server. The result string returned by the password server contains additional information about the error.

User response

Choose a new password that meets the password policy. Contact your service representative if the error persists.

EUVF02043E Password change failed.

Explanation

The password change request was not successful.

User response

Contact your service representative if the error persists.

EUVF02044I Password changed.

Explanation

The password change request was successful.

User response

None

EUVF02047W Unable to set the requested FIPS level *level*. The process will continue to run in the current FIPS level *level*.

Explanation

The current process FIPS level can not be set to the value specified for FIPS level. The level remains unchanged. For more information on the FIPS level values, see “Configuration of encryption types and FIPS level” on page 17.

User response

Ensure that the FIPS level for the process has not been set prior to calling the first Kerberos API, or disable the setting of the FIPS level in the Kerberos configuration file (fipslevel = -1).

Security server messages (numbers EUVF04000 - EUVF05999)

EUVF04001I **Security server version *version.release* Service level *level*.**

Explanation

The security server is starting. This message displays the version, release, and service level of the security server.

User response

None

EUVF04002I **Security runtime version *version.release* Service level *level*.**

Explanation

The security server is starting. This message displays the version, release, and service level of the runtime DLL.

User response

None

EUVF04003A **Unable to make address space non-swappable: Error *error-code*.**

Explanation

The security server is unable to make its address space non-swappable. The error code is the value returned by the SYSEVENT system service. An error code of 1 indicates the security server job step is not APF-authorized. Refer to the description of the SYSEVENT macro in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services Reference*, for more information on the error.

User response

Verify that the SKRBKDC started task is APF-authorized. Contact your service representative if the error persists.

EUVF04004E **The *function-name* system function detects an error: *error-message*.**

Explanation

A system function detected an error. The error message text is returned by the **strerror()** function. Refer to the description of the failing system function in *z/OS C/C++ Run-Time Library Reference*, SA22-7821-03, for more information on the error.

User response

Contact your service representative if the error persists.

EUVF04005E **Insufficient storage available.**

Explanation

The security server is unable to obtain storage for an internal control block.

User response

Increase the storage available to the program and then retry the request.

EUVF04006I **Security server shutdown requested.**

Explanation

The system operator has entered a STOP command for the security server.

User response

None

EUVF04007E **Unrecognized security server command: Specify DISABLE, DISPLAY, ENABLE, PROP or DEBUG.**

Explanation

An unrecognized command name is specified on a MODIFY operator command. The only valid security server commands are DISABLE, DISPLAY, ENABLE, PROP or DEBUG.

User response

Specify a valid security server command.

EUVF04008I **Debug option processed: *debug-option*.**

Explanation

The indicated debug request has been processed by the security server.

User response

None

EUVF04009E Incorrect command option specified.

Explanation

An incorrect security server command option was specified. The valid DISPLAY command options are:

- ADMIN - Display the current status of the Kerberos administration service
- CREDS - Display credentials data space allocations
- CRYPTO - Display the available encryption types.
- LEVEL - Display the security server version, release, and service level.
- NETWORK - Display the network interface status.
- PROP - Display the database propagation status.
- XCF - Display security server sysplex status

The valid DEBUG command options are:

- OFF - Turn off debug messages.
- ON - Turn on debug messages.
- *subcomp.level,subcomp.level,...* - Set the debug level for one or more subcomponents.

User response

Specify a valid command option.

EUVF04010A Database type *type* is not supported.

Explanation

The SKDC_DATABASE environment variable specifies an unsupported database type.

User response

Specify a supported database type. The security server supports the SAF and NDBM databases.

EUVF04011E Unable to receive datagram. *error_text*.

Explanation

The security server is unable to receive a datagram from the network. The error text is returned by the **strerror()** routine.

User response

Contact your network support group.

EUVF04012E Unable to send response to *network-address*. *error-text*.

Explanation

The security server is unable to send data to the specified network address. The error text is returned by the **strerror()** routine.

User response

Contact your network support group.

EUVF04013E Unable to initialize local services: Error *error-code*, Reason *reason-code*.

Explanation

The security server is unable to initialize the local services support. This support is used by applications running on the same system as the security server. The error code indicates the failing system function and the reason code is the error code returned by the system function.

The following error codes are defined:

- 1 = The job step is not APF-authorized.
- 2 = The security server is already running.
- 3 = ESTAEX failed. See the ESTAEX macro description in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services* for more information on the error.
- 5 = LXRES failed. See the LXRES macro description in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services* for more information.
- 6 = ETCRE failed. See the ETCRE macro description in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services* for more information.
- 7 = ETCON failed. See the ETCON macro description in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services* for more information.
- 8 = IEANTCR failed. See the IEANTCR macro description in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services* for more information.
- 9 = CTRACE DEFINE failed. See the CTRACE macro description in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services* for more information.

User response

Contact your service representative, if you are unable to correct the error.

EUVF04014E **Unable to end local services: Error *error-code*, Reason *reason-code*.**

Explanation

The security server is unable to end the local services support. The error code indicates the failing system function and the reason code is the error code returned by the system function.

The following error codes are defined:

- 102 = Unable to cancel ESTAEX. See the ESTAEX macro description in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services* for more information on the error.
- 110 = IEANTDL failed. See the IEANTDL macro description in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services* for more information.
- 111 = Unable to obtain control lock.
- 112 = CTRACE DELETE failed. See the CTRACE macro description in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services* for more information.

User response

Contact your service representative.

EUVF04015E **Local program call request failed: Error *error-code*.**

Explanation

The security server is unable to process a local program call request.

The following error codes are defined:

- 8 = Parameter buffer overflow
- 12 = Unable to allocate storage
- 16 = Local service support not enabled
- 20 = Program call task abended
- 24 = Unable to obtain control lock
- 28 = SRB mode is not supported.

User response

Contact your service representative.

EUVF04016E **Login audit mode *mode* is not supported.**

Explanation

The SKDC_LOGIN_AUDIT environment variable specified an unsupported audit mode. The supported audit modes are NONE, FAILURE, and ALL.

User response

Specify a supported login audit mode.

EUVF04017A **Unable to initialize the KDC.**

Explanation

The security server is unable to initialize the KDC (Key Distribution Center). A previous error message contains more information about the failure.

User response

Correct the problem and then restart the security server. Contact your service representative if the error persists

EUVF04018I **Security server initialization complete.**

Explanation

The security server initialization is complete.

User response

None

EUVF04019A **Realm name *name* contains separator characters.**

Explanation

The realm name contains the '/' or '@' characters. These characters are used as component separators and may not be used in the realm name.

User response

Use a valid realm name.

EUVF04020A **Security server is already running.**

Explanation

Security server is already running.

User response

None

EUVF04021A **Unable to initialize registry database support.**

Explanation

The security server is unable to initialize the registry database support. A previous message provides more information on the cause of the failure.

User response

Correct the problem and restart the security server. Contact your service representative if the error persists.

EUVF04022I Security server start command processed.

Explanation

The security server has completed processing the START command.

User response

None

EUVF04023I Security server stop command received.

Explanation

The security server has received a STOP command.

User response

None

EUVF04024E Security server is not available.

Explanation

The component trace command processor is unable to call the Kerberos security server because the security server is either not running or is in the process of stopping.

User response

Restart the Kerberos security server and then retry the TRACE command.

EUVF04025E Unable to call security server: Error *number*, Reason *code*.

Explanation

The component trace command processor is unable to call the Kerberos security server due to an error on the program call request.

The error codes have the following values:

- 8 = Parameter buffer overflow.
- 12 = Unable to allocate storage.
- 16 = Local service support not enabled.
- 20 = Program call task abended.
- 24 = Unable to obtain control lock.
- 28 = SRB mode is not supported.

User response

Contact your service representative if the error persists.

EUVF04026E Incorrect trace option specified.

Explanation

The OPTIONS parameter on the TRACE CT command does not specify a valid list of subcomponent trace levels.

The OPTIONS parameter specifies the list of subcomponent trace levels as OPTIONS=(subcomp1.lvl1,subcomp2.lvl2,...). Trace messages for a particular subcomponent will not be logged unless the subcomponent is included in the trace list and the message level is greater than or equal to the specified level. An asterisk (*) may be used to specify all subcomponents. Trace level 1 generates the minimum amount of trace message output, trace level 8 generates the maximum amount of trace message output, and trace level 9 generates data dumps in addition to the trace messages. The subcomponent list consists of a subcomponent name and a trace level separated by a period. Multiple subcomponents may be specified by separating the entries with commas.

User response

Specify a valid list of subcomponent trace levels.

EUVF04027E The trace buffer size must be between 64K and 512K.

Explanation

The trace buffer size specified on the TRACE CT command must be between 64K and 512K.

User response

Specify a valid trace buffer size.

EUVF04028E Unable to retrieve information from the System Authorization Facility registry. SAF error *error-code*, Return code *return-code*, Reason code *reason-code*.

Explanation

The call to the IRRSIM00 system function fails with the indicated error code. Refer to *z/OS Security Server RACF Callable Services* for more information on the IRRSIM00 callable service and its error return values.

User response

Contact your service representative if the error persists.

EUVF04029E **Kerberos segment field *field-name* for user *user-name* is not valid.**

Explanation

The Kerberos segment for the specified user is not valid. The data is either too long or is not formatted correctly.

User response

Use the ALTUSER (or equivalent) command to correct the Kerberos segment data.

EUVF04030E **Registry realm *name* does not match configured realm *name*.**

Explanation

The realm name in the security registry is not the same as the default realm specified in the krb5.conf configuration file.

User response

Change the default realm name or create a new registry for the realm.

EUVF04031E **Limit of *number* sockets exceeded.**

Explanation

The maximum number of open sockets has been exceeded.

User response

Run additional instances of the security server on other systems within the Kerberos realm.

EUVF04032E **Unable to receive data from *network-address*. *error-text*.**

Explanation

The security server is unable to receive data from a client at the specified network address. The error text is returned by the **strerror()** routine.

User response

Contact your network support group.

EUVF04033E **Message received from *network-address* with length *size* exceeds the maximum size.**

Explanation

The security server received a request that is too large to be processed. The maximum request message size is 32768.

User response

Verify that the client is sending a valid Kerberos request to the security server. Contact your service representative if the problem persists.

EUVF04034E **R_kerbinfo request fails: Function *code*, Error *error-code*, RC *return-code*, Reason *reason-code*. Kerberos name: *name***

Explanation

The **R_kerbinfo** (IRRSMK00) function call failed. Refer to *z/OS Security Server RACF Callable Services* for more information. In addition to the return codes documented in the callable services publication, you can also receive error code 8, return code 8, reason code 16, if the SKRBKDC&tab; started task is not APF-authorized.

User response

Contact your service representative if you are unable to correct the problem.

EUVF04035E **Local Kerberos realm is not defined.**

Explanation

The local Kerberos realm has not been defined in the system security database. Refer to *z/OS Security Server RACF Command Language Reference* for more information.

User response

Define the local realm and then restart the security server.

EUVF04036E **Unable to generate PassTicket for *userid*.**

Explanation

The security server is unable to generate a PassTicket for the indicated user.

User response

Verify that the PTKTDATA class is active and that the SKRBKDC application has been defined in the PTKTDATA class with a valid secured signon key.

EUVF04037E **Unable to change password for *userid*. SAF error *code*, Return code *code*, Reason code *code*.**

Explanation

The security server is unable to change the password for the indicated user. Refer to *z/OS Security Server RACROUTE Macro Reference* for more information about the error codes returned by the RACROUTE REQUEST=VERIFY function.

User response

Contact your service representative if the error persists.

EUVF04038I **Kerberos login successful for *principal* at *address*.**

Explanation

A request for an initial ticket was successful. This message is controlled by the SKDC_LOGIN_AUDIT environment variable.

User response

None

EUVF04039W **Kerberos login failed for *principal* at *address*. KDC status code: *error-text*.**

Explanation

A request for an initial ticket failed. This message is controlled by the SKDC_LOGIN_AUDIT environment variable.

User response

Regenerate the key for the principal associated with the userid. Additionally, contact the user attempting to get the initial ticket if the problem persists.

EUVF04040E **Unable to encode database entry for *name*: Status *status-code - status-message***

Explanation

The Kerberos security server was unable to encode a database entry.

User response

Contact your service representative.

EUVF04041E **Unable to read database master key: *error-text.***

Explanation

The Kerberos security server was unable to read the database master key from the stash file. This key is used to encrypt entries in the principal database. The error text is returned by the **strerror()** routine.

User response

Verify that the stash file **/var/skrb/krb5kdc/.k5** exists and is accessible. Contact your service representative if the error persists.

EUVF04042E **Unable to write database master key: *error-text.***

Explanation

The Kerberos security server was unable to write the database master key to the stash file. This key is used to encrypt entries in the principal database. The error text is returned by the **strerror()** routine.

User response

Verify that the stash file **/var/skrb/krb5kdc/.k5** exists and is accessible. Contact your service representative if the error persists.

EUVF04043E **Unable to store database entry for *name*. *Error-text.***

Explanation

The Kerberos security server was unable to store the indicated entry in its database. The database files are stored in the **/var/skrb/krb5kdc** directory. The error text is returned by the **strerror()** routine.

User response

Verify that the **/var/skrb/krb5kdc** directory exists and is mounted in read/write mode. Contact your service representative if the error persists.

EUVF04044E **Unable to fetch database entry for *name*. *Error-text.***

Explanation

The Kerberos security server was unable to retrieve the indicated entry in its database. The database files are stored in the **/var/skrb/krb5kdc** directory. The error text is returned by the **strerror()** routine.

User response

Verify that the **/var/skrb/krb5kdc** directory exists and is mounted in read/write mode. Contact your service representative if the error persists.

EUVF04046E **Unable to decode RPC message received from *network-address.***

Explanation

The Kerberos administration server was unable to decode an RPC message received from a client.

User response

Contact your service representative if the error persists.

EUVF04047E **Unable to create directory *name*.**

Explanation

The Kerberos security server was unable to create the indicated directory.

User response

Verify that all path components exist and that the UID of the security server has permission to create the directory specified in the message. Contact your service representative if the error persists.

EUVF04048E **Unable to open database *name*. *Error-text.***

Explanation

The Kerberos security server was unable to open the indicated registry database. The error text is returned by the **strerror()** routine.

User response

Verify that all path components exist and that the UID of the security server has permission to open the database file specified in the message. Contact your service representative if the error persists.

EUVF04049E **Unable to open administration key table *name*. Status *status-code - status-message.***

Explanation

The Kerberos security server was unable to open the administration key table.

User response

Verify that all path components exist and that the UID of the security server has permission to open the key table file specified in the message. Contact your service representative if the error persists.

EUVF04050E **Unable to add entry to administration key table *name*. Status *status-code - status-message.***

Explanation

The Kerberos security server was unable to add an entry to the administration key table.

User response

Verify that all path components exist and that the security server is running with UID 0. Contact your service representative if the error persists.

EUVF04051E **Unable to read entry for principal *name*. Status *status-code - status-message.***

Explanation

The Kerberos security server was unable to read a principal entry from the database.

User response

Verify that the principal entry exists in the database. Contact your service representative if the error persists.

EUVF04052E **GSS-API function *name* detects error. *major-status minor-status***

Explanation

The Kerberos security server was unable to process an administration request due to a GSS-API error. The z/OS Security Server supports the Kerberos Administration Version 2 protocol as implemented in MIT Kerberos 1.2.2. Refer to the description for the failing GSS-API function in *z/OS Integrated Security Services Network Authentication Service Programming* for more information on the error.

User response

Contact your service representative if the error persists.

EUVF04053E Unable to create credentials data space: Error *error-code*, Reason *reason-code*.

Explanation

The Kerberos security server was unable to create the credentials data space.

The error codes have the following values:

- 1 = DSPSERV CREATE failed. The reason code contains the DSPSERV return code in the upper halfword and bits 8-23 of the DSPSERV reason code in the lower halfword. See the description of the DSPSERV macro in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services Reference* for more information on the error.
- 2 = ALESERV ADD failed. The reason code is the ALESERV return code. See the description of the ALESERV macro in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services Reference* for more information on the error.

User response

Security server initialization continues but data space services are not available. Contact your service representative if the error persists.

EUVF04054E Unable to delete credentials data space: Error *error-code*, Reason *reason-code*.

Explanation

The Kerberos security server was unable to delete the credentials data space.

The error codes have the following values:

- 1 = DSPSERV DELETE failed. The reason code contains the DSPSERV return code in the upper halfword and bits 8-23 of the DSPSERV reason code in the lower halfword. See the description of the DSPSERV macro in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services Reference* for more information on the error.
- 2 = ALESERV DELETE failed. The reason code is the ALESERV return code. See the description of the ALESERV macro in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services Reference* for more information on the error.

User response

Security server termination continues. Contact your service representative if the error persists.

EUVF04055E Unable to extend the credentials data space: Error *error-code*, Reason *reason-code*.

Explanation

The Kerberos security server is unable to increase the size of the credentials data space.

The error codes have the following values:

- 1 = DSPSERV EXTEND failed. The reason code contains the DSPSERV return code in the upper halfword and bits 8-23 of the DSPSERV reason code in the lower halfword. See the description of the DSPSERV macro in the appropriate volume of *z/OS MVS Programming: Authorized Assembler Services Reference* for more information on the error.

User response

The new credentials are not stored in the credentials data space. The SKDC_CREDS_SIZE environment variable specifies the maximum allowable size for the credentials data space.

EUVF04056E Unable to initialize cross-system services: Error *error-code*, Reason *reason-code*.

Explanation

The Kerberos security server is unable to initialize cross-system services.

The error codes have the following values:

- 1 = The job step is not APF-authorized.
- 3 = IXCJOIN failed. The reason code contains the IXCJOIN return code in the upper halfword and the IXCJOIN reason code in the lower halfword. See the description of the IXCJOIN macro in *z/OS MVS Programming: Sysplex Services Reference* for more information on the error.
- 4 = IXCQUERY failed. The reason code contains the IXCQUERY return code in the upper halfword and the IXCQUERY reason code in the lower halfword. See the description of the IXCQUERY macro in *z/OS MVS Programming: Sysplex Services Reference* for more information on the error.

User response

Security server initialization continues but cross-system services are not available. Contact your service representative if the error persists.

EUVF04057E Unable to end cross-system services: Error *error-code*, *reason-code*.

Explanation

The Kerberos security server is unable to end cross-system services.

The error codes have the following values:

- 5 = IXCLEAVE failed. The reason code contains the IXCLEAVE return code in the upper halfword and the IXCLEAVE reason code in the lower halfword. See the description of the IXCLEAVE macro in *z/OS MVS Programming: Sysplex Services Reference* for more information on the error.

User response

Security server processing continues. Contact your service representative if the error persists.

EUVF04058I System *name* has joined the Kerberos security server group.

Explanation

The SKRBKDC started task has completed initialization on the indicated system and is now a member of the EUVFSKRB cross-system group.

User response

None

EUVF04059I System *name* has left the Kerberos security server group.

Explanation

The SKRBKDC started task is stopping on the indicated system.

User response

None

EUVF04060I Cross-system services ended due to sysplex partitioning.

Explanation

The local system is leaving the sysplex. As a result, Kerberos security server cross-system services are no longer available.

User response

Security server processing continues.

EUVF04061E Unable to send cross-system message: Error *error-code*, Reason *reason-code*.

Explanation

The Kerberos security server is unable to send a message to another member of the Kerberos security server group.

The error codes have the following values:

- 1 = Unable to obtain XCF control lock on target system.
- 2 = Cross-system services are not available.
- 3 = Requested token not found on target system.
- 4 = User not authorized to access token data.
- 5 = Unable to allocate storage on the target system.
- 6 = Target replica is not a member of the security server group.
- 7 = Target replica is not active.
- 8 = IXCMSGO failed. The reason code contains the IXCMSGO return code in the upper halfword and the IXCMSGO reason code in the lower halfword. See the description of the IXCMSGO macro in *z/OS MVS Programming: Sysplex Services Reference* for more information on the error.
- 9 = IXCMSGI failed on the target system. The reason code contains the IXCMSGI return code in the upper halfword and the IXCMSGI reason code in the lower halfword. See the description of the IXCMSGI macro in *z/OS MVS Programming: Sysplex Services Reference* for more information on the error.
- 10 = Request function code is not supported.
- 11 = Request canceled.
- 12 = Unknown notification message.
- 13 = No response received from target system.
- 14 = Unable to allocate storage on the local system.
- 15 = IXCMSGI failed on the local system. The reason code contains the IXCMSGI return code in the upper halfword and the IXCMSGI reason code in the lower halfword. See the description of the IXCMSGI macro in *z/OS MVS Programming: Sysplex Services Reference* for more information on the error.

User response

The request was not processed. Contact your service representative if the error persists.

EUVF04062A The security server is not APF-authorized.

Explanation

The Kerberos security server must be APF-authorized.

User response

Verify that the dataset containing the EUVFSKDC load module is APF-authorized. If you are using a STEPLIB or JOBLIB for the SKRBKDC started task, verify that all datasets in the concatenation are APF-authorized.

EUVF04063E Unable to allocate *size* bytes in the credentials data space.

Explanation

The credentials data space is full. The SKDC_CREDS_SIZE environment variable can be used to increase the size of the credentials data space. SKDC_CREDS_SIZE specifies the credentials data space size in kilobytes, with a minimum value of 1024, a maximum value of 2097148, and a default value of 20480.

User response

Increase the size of the credentials data space and then restart the Kerberos security server.

EUVF04064I Sysplex status.

Explanation

This message is displayed in response to the Kerberos security server DISPLAY XCF command. The remaining lines in this multi-line message display the status of each Kerberos security server in the sysplex. A security server is ACTIVE if the SKRBKDC started task is running. A security server is INACTIVE if the SKRBKDC started task has been stopped. No entry is displayed for a system where the SKRBKDC started task has not been active at any time since the local security server was started.

User response

None

EUVF04065I No active security servers.

Explanation

There are no active security servers in the sysplex. This message can occur if there was an error in setting up the cross-system coupling facility support.

User response

None

EUVF04066I Data space status.

Explanation

This message is displayed in response to the Kerberos security server DISPLAY CREDS command. The remaining lines in this multi-line message display the data space allocations.

User response

None

EUVF04067I No data space allocations.

Explanation

There are no data space allocations.

User response

None

EUVF04068I Maximum number of lines displayed.

Explanation

A maximum of 253 status lines can be displayed in response to a single command.

User response

None

EUVF04069I Listening for requests on network interface *address*.

Explanation

The security server is listening for requests on the indicated network interface.

User response

None

EUVF04070I No longer listening for requests on network interface *address*.

Explanation

The security server is no longer listening for requests on the indicated network interface.

User response

None

EUVF04071E **Unable to bind socket to *network-address: error-text*.**

Explanation

The security server is unable to bind a socket to listen for client requests on the indicated network interface. The error text is returned by the **strerror()** routine.

User response

Ensure that the TCP/IP configuration profile does not reserve the network port for use by another application. The SKDC_PORT, SKDC_KPASSWD_PORT, and SKDC_KADMIN_PORT environment variables can be used to change the ports used by the security server.

EUVF04072I **Security server restart registration complete on *system*.**

Explanation

The security server has successfully registered with ARM (Automatic Restart Management) on the indicated system. The security server is automatically restarted if it fails unexpectedly (the security server is not restarted if it detects an error and stops). The ARM element type is SYSKERB and the ARM element name is EUVFKDC_system-name. The ARM policy can be used to override the default registration values if needed. Refer to *z/OS MVS Programming: Sysplex Services Guide* for more information on automatic restart management.

User response

None

EUVF04073I **Security server restarting on *system*.**

Explanation

The security server is being restarted following an unexpected failure. The RESTART_ATTEMPTS value in the ARM policy determines the number of restarts attempted. Refer to *z/OS MVS Programming: Sysplex Services Guide* for more information on automatic restart management.

User response

None

EUVF04074E **Unable to register for restart: Error *error-code*, Reason *reason-code*.**

Explanation

The security server is unable to register with ARM (Automatic Restart Management). The IXCARM request failed with the indicated error and reason codes. See the description of the IXCARM macro in *z/OS MVS Programming: Sysplex Services Reference* for more information on the error.

User response

Contact your service representative if you are unable to correct the error.

EUVF04075E **Cryptographic status.**

Explanation

This message is displayed in response to the Kerberos security server DISPLAY CRYPTO command. The remaining lines in this multi-line message display the available encryption types and whether encryption and decryption operations are performed using the Integrated Cryptographic Service Facility (ICSF).

User response

None

EUVF04076E **Unable to open access control file *filename: error-text***

Explanation

The Kerberos security server was unable to open an access control file. The security server uses this file to control access to the Kerberos functions. The error text is returned by the **strerror()** routine.

User response

Verify that all path components exist, the access control file exists, and that the UID of the security server has permission to open the access control file specified in the message. Contact your service representative if the error persists.

EUVF04077E **Unable to read access control file *filename: error-text***

Explanation

The Kerberos security server is unable to read an access control file. The error text is returned by the **strerror()** routine.

User response

Contact your service representative if the error persists.

EUVF04078E Unable to convert *expression* to a regular expression: *error-text*

Explanation

The Kerberos security server is unable to parse a line in the administration access control file. The expression cannot be converted to a regular expression for the indicated reason. The error text is returned by the **strerror()** routine.

User response

Correct the expression in the administration access control file. Contact your service representative if the error persists.

EUVF04079E Unable to decode database entry for *name*: Status *status-code* - *status-message*

Explanation

The Kerberos security server is unable to decode a database entry.

User response

Contact your service representative.

EUVF04080E Unable to delete database entry for *name*: *error-message*

Explanation

The Kerberos security server is unable to delete an entry from the registry database.

User response

Contact your service representative if the error persists.

EUVF04081E Unable to validate database master key: Status *status-code* - *status-message*

Explanation

The Kerberos security server is unable to validate the database master key. This error can occur if the master key stash file is generated using the wrong database password. This error also can occur when loading a new database if the wrong master key is entered in response to the database password prompt.

User response

For an existing database, create a master key stash file containing the correct database master key. For a new database, enter the correct database master key when prompted. Contact your service representative if the error persists.

EUVF04082E The *option-name* option requires a value.

Explanation

A command option is specified without a corresponding value.

User response

Specify a value for the indicated option.

EUVF04083E *option-name* is not a valid command option.

Explanation

An unrecognized command option is specified.

User response

Specify a valid command option.

EUVF04084E The *option-name* option is not valid for the '*function-name*' function.

Explanation

A valid option is specified but the option is not valid for the requested function.

User response

Specify a valid command option.

EUVF04085I kdb5_ndbm function [-e keytypes] [-k keytype] [-mkey_convert] [-hkey_convert] [-compat] [-v] filename

Explanation

This message displays the command syntax for the **kdb5_ndbm** command.

User response

None

EUVF04086E **No database utility function specified.**

Explanation

A database utility command is entered without specifying a function to be performed.

User response

Specify a database utility function.

EUVF04087E ***function-name* is not a valid database utility function.**

Explanation

An unrecognized function is specified for a database utility command.

User response

An unrecognized function is specified for a database utility command.

EUVF04088E **Unable to create KDC database: Status *status-code - status-message***

Explanation

The Kerberos security server is unable to create the registry database.

User response

Ensure that the failing command is being run under a user ID with write access to the **/var/skrb/krb5kdc** directory and to all of the files in that directory. Contact your service representative if the error persists.

EUVF04089E **Unable to obtain the default realm: Status *status-code - status-message***

Explanation

The **krb5_get_default_realm()** function failed.

User response

Verify that the **/etc/skrb/krb5.conf** configuration file exists and contains an entry for the default realm. Contact your service representative if the error persists.

EUVF04090E **Encryption type *name* is not supported.**

Explanation

An unsupported encryption type was specified.

User response

Specify a supported encryption type.

EUVF04091R **Enter the KDC database master password:**

Explanation

Enter the master password for the KDC database. The password should not be obvious or easily guessed since it will be used to generate the master key for the database. The master key is used to encrypt the database entries.

User response

Enter the password string for the database master key.

EUVF04092R **Re-enter the KDC database master password:**

Explanation

Re-enter the master password for the KDC database to verify the password was entered correctly.

User response

Enter the same password string as you entered for the initial prompt.

EUVF04093E **Unable to read the KDC database master password: Status *status-code - status-message***

Explanation

The Kerberos security server is unable to prompt for the database master password.

User response

Contact your service representative if the error persists.

EUVF04094E Unable to generate the KDC database master key: Status *status-code - status-message*

Explanation

The Kerberos security server is unable to convert the password string into an encryption key.

User response

Contact your service representative.

EUVF04095I KDC database master key created.

Explanation

The KDC database master key has been created.

User response

None

EUVF04096E A KDC database already exists.

Explanation

A request to create a new KDC database cannot be completed because a database already exists.

User response

Use the **kdb5_ndbm** command to remove the existing database and then retry the failing command.

EUVF04097E Unable to delete database file *filename*: *error-text*

Explanation

The Kerberos security server is unable to delete a database file. The error text is returned by the **strerror()** routine.

User response

Verify that the command is being run by a user ID with UID 0. Contact your service representative if the error persists.

EUVF04098I KDC database files deleted.

Explanation

The KDC database files have been deleted.

User response

None

EUVF04099I KDC database created.

Explanation

The KDC database has been created.

User response

None

EUVF04100E The dump filename must be specified.

Explanation

A database dump or load function was requested but no dump filename is provided.

User response

Specify the dump filename.

EUVF04101E Unable to dump the KDC database: Status *status-code - status-message*

Explanation

The Kerberos security server is unable to dump the KDC database.

User response

Contact your service representative if the error persists.

EUVF04102I KDC database dump file *filename* created.

Explanation

The KDC database has been dumped to the indicated file.

User response

None

EUVF04103E Unable to load the KDC database: Status *status-code - status-message*

Explanation

The Kerberos security server is unable to load the KDC database.

User response

Contact your service representative.

EUVF04104I KDC database loaded from file *filename*.

Explanation

The KDC database has been loaded from the indicated file.

User response

None

EUVF04105E The KDC database does not exist.

Explanation

An attempt to read from the KDC database failed because the database does not exist.

User response

Verify that the database files exist. The files are located in the **/var/skrb/krb5kdc** directory. Contact your service representative if the error persists.

EUVF04106R Enter the database dump password:

Explanation

Enter the password for the database dump. The password should not be obvious or easily guessed since it will be used to encrypt the database entries in the dump.

User response

Enter the password string for the database dump key.

EUVF04107R Re-enter the database dump password:

Explanation

Re-enter the password for the database dump to verify the password was entered correctly.

User response

Enter the same password string as you entered for the initial prompt.

EUVF04108E Unable to read the database dump password: Status *status-code* - *status-message*

Explanation

The Kerberos security server is unable to prompt for the database dump password.

User response

Contact your service representative if the error persists.

EUVF04109E Unable to generate the database dump key: Status *status-code* - *status-message*

Explanation

The Kerberos security server is unable to convert the password string into an encryption key.

User response

Contact your service representative.

EUVF04110E Unable to write to dump file filename: *error-text*

Explanation

The Kerberos security server is unable to write to the indicated database dump file.The error text is returned by the **strerror()** routine.

User response

Verify that the user has write access to the directory and the file. Contact your service representative if the error persists.

EUVF04111E Unable to read from dump file filename: *error-text*

Explanation

The Kerberos security server is unable to read from the indicated database dump file. The error text is returned by the **strerror()** routine.

User response

Verify that the user has read access to the directory and the file. Contact your service representative if the error persists.

EUVF04112E Principal *name* references unknown policy *name*.

Explanation

A principal entry in the Kerberos database contains a reference to an unknown policy. This indicates the database has become corrupted.

User response

Either restore the database from a backup or remove the policy reference from the principal entry. Contact your service representative if the error persists.

EUVF04113E **Database entry *name* with size *number* exceeds maximum size *number*.**

Explanation

A database entry is larger than the maximum supported size. This indicates the database has become corrupted.

User response

Restore the database from a backup. Contact your service representative if the error persists.

EUVF04114E **Architected principal *name* is not found in the KDC database.**

Explanation

The Kerberos security server is unable to locate a required principal in the database. This indicates the database has become corrupted.

User response

Restore the database from a backup. Contact your service representative if the error persists.

EUVF04115W **Password history for *principal* with *number* entries is too large.**

Explanation

The password history for the indicated principal has become too large to be stored in the Kerberos database. The oldest entries will be removed until the resulting history is small enough to fit in a database record.

User response

Reduce the history count for the policy associated with the principal.

EUVF04116E **File *filename* does not contain a valid database dump.**

Explanation

The indicated file does not contain a valid Kerberos database dump. This problem can occur if the dump is created using the database utility provided with an older Kerberos implementation.

User response

Recreate the dump using the current level of the Kerberos database utility command.

EUVF04117E **Dump record type *name* is not valid.**

Explanation

A record in the database dump file was not recognized.

User response

Contact your support representative.

EUVF04118E **Unable to rename database file from *oldname* to *newname*: *error-text***

Explanation

The Kerberos security server is unable to rename a database file. The error text is returned by the **strerror()** routine.

User response

Contact your service representative.

EUVF04119E **No history key is available for master key type *encryption-type*.**

Explanation

The history principal (**kadmin/history**) does not have a key for the encryption type defined by the database master key.

User response

Change the keys for the history principal and generate a new key with the same encryption type as the database master key.

EUVF04120E **Unable to encrypt database entry for *name*: Status *status-code* - *status-message***

Explanation

The Kerberos security server was unable to encrypt an entry in the KDC database.

User response

Contact your service representative.

EUVF04121E Unable to decrypt database entry for *name*: Status *status-code - status-message*

Explanation

The Kerberos security server is unable to decrypt an entry in the KDC database. This error can be caused by an incorrect database master key.

User response

For a secondary KDC, destroy the existing Kerberos database and then recreate it from the primary KDC database. Ensure that the correct database master key is entered when you are prompted for the master password. Contact your service representative if the error persists.

EUVF04122E Principal name *name* is not valid.

Explanation

A principal name is not composed of valid graphical characters as determined by the current locale. In addition, the backslash and commercial at-sign characters are not allowed in a principal name.

User response

Use only valid graphical characters in principal names.

EUVF04123E Policy name *name* is not valid.

Explanation

A policy name is not composed of valid graphical characters as determined by the current locale. In addition, the backslash character is not allowed in a policy name.

User response

Use only valid graphical characters in principal names.

EUVF04124E Unrecognized propagation role specified for *name*.

Explanation

The **kpropd.acl** access control file contains an unrecognized role specified for the indicated server entry. The valid roles are Primary, Replace, Compat, and Update.

User response

Specify a valid propagation role.

EUVF04125E Unable to resolve host principal for *name*: Status *status-code - status-message*

Explanation

The Kerberos security server is unable to convert a host name to a Kerberos principal. This error can occur if the host name is not defined in the DNS name server or the DNS name server cannot be reached.

User response

Ensure the host name is defined and the DNS name server can be reached.

EUVF04126E Unable to log *type* request from *name*: Status *status-code - status-message*

Explanation

The Kerberos security server is unable to log an administration request from the indicated user. This means that the database update will not be propagated to secondary security servers that are using the update protocol. The change has been made to the database on the primary security server.

User response

Use the Kerberos security server PROP command to force a full database replication for each secondary security server. Contact your service representative if the error persists.

EUVF04127E Database propagation failed to *host*: Status *status-code - status-message*

Explanation

The Kerberos security server is unable to send a database update to the indicated secondary server.

User response

Verify that the secondary server is running and that there are no network problems. The update will be retried at the next propagation interval or the PROP command can be used to initiate a manual replication. Contact your service representative if the error persists.

EUVF04128E Unable to fetch update *number*: *error-text*

Explanation

The Kerberos security server is unable to retrieve the indicated update from its database. The database files are stored in the **/var/skrb/krb5kdc** directory. The error text is returned by the **strerror()** routine.

User response

Verify that the **/var/skrb/krb5kdc** directory exists and is mounted in read/write mode. Contact your service representative if the error persists.

EUVF04129E **Unable to store update *number*: *error-text***

Explanation

The Kerberos security server is unable to store the indicated update in its database. The database files are stored in the **/var/skrb/krb5kdc** directory. The error text is returned by the **strerror()** routine.

User response

Verify that the **/var/skrb/krb5kdc** directory exists and is mounted in read/write mode. Contact your service representative if the error persists.

EUVF04130E **Update to delete update *number* : *error-text***

Explanation

The Kerberos security server is unable to delete the indicated update from its database. The database files are stored in the **/var/skrb/krb5kdc** directory.The error text is returned by the **strerror()** routine.

User response

Verify that the **/var/skrb/krb5kdc** directory exists and is mounted in read/write mode. Contact your service representative if the error persists.

EUVF04131I **Propagation status.**

Explanation

This message is displayed in response to the Kerberos security server DISPLAY PROP command. The remaining lines in this multi-line message display the propagation status for each security server in the realm.

User response

None

EUVF04132I **Propagation complete.**

Explanation

The Kerberos security server has successfully completed a database propagation request.

User response

None

EUVF04133E **Propagation failed: Status *status-code - status-message***

Explanation

The Kerberos security server is unable to process a PROP command for the indicated reason. The status of the secondary KDC involved in the propagation remains unchanged.

User response

Ensure that the secondary KDC is running and that there are no network problems. Contact your service representative if the error persists.

EUVF04134E **Missing command option.**

Explanation

A Kerberos security server command was entered that requires a command option but no command option was entered.

User response

Enter a complete Kerberos security server command.

EUVF04135I **No propagation status.**

Explanation

This message is displayed in response to the Kerberos security server DISPLAY PROP command when database propagation is not enabled or the security server is a secondary security server.

User response

Enter the DISPLAY PROP command at the primary security server for the realm.

EUVF04136E **The PROP command is not available.**

Explanation

The Kerberos security server PROP command was entered on a system that does not support database propagation. This can occur if the Kerberos database does not support propagation, propagation is not

enabled, or the Kerberos security server is not the primary security server for the realm.

User response

None

EUVF04137E The primary security server cannot be changed.

Explanation

The propagation control file has been updated to change the role of the local Kerberos security server from primary to secondary or from secondary to primary while the security server is running. The security server must be stopped and then restarted in order to change its role.

User response

Stop both security servers involved in the role change, propagate the latest version of the Kerberos database from the old primary system to the new primary system, and then restart both security servers.

EUVF04138E Unable to receive propagation from *host* - Status *status-code* - *status-message*

Explanation

The Kerberos security server was unable to receive a database propagation from the indicated host.

User response

Verify that the primary security server is correctly identified in the **kpropd.acl** configuration file and that there are no network errors. The primary security server must have a DNS entry and the entry must be associated with the IP address used for the database propagation. Contact your service representative if the error persists.

EUVF04139E Administration services are not available.

Explanation

The Kerberos security server database does not support the administration functions. Database administration must be performed using the system security commands.

User response

None

EUVF04140I Administration services are enabled.

Explanation

Kerberos administration services are enabled.

User response

None

EUVF04141I Administration services are disabled.

Explanation

Kerberos administration services are disabled. Either the Kerberos database does not support the administration functions or the DISABLE ADMIN command has been issued.

User response

Use the Kerberos security server ENABLE ADMIN command to enable administration services if the Kerberos database supports the administration functions.

EUVF04142I kpropd [-r realm] [-P port] [-v]

Explanation

This message lists the command syntax for the kpropd command.

User response

None

EUVF04143E *port* is not a valid port specification.

Explanation

The port specification is not a decimal number between 1 and 65535.

User response

Specify a valid port.

EUVF04144E No propagation servers are defined.

Explanation

No propagation servers are defined in the **/etc/skrb/home/kdc/kpropd.acl** configuration file. A database propagation is accepted only from servers listed in this configuration file.

User response

Add the name of the primary KDC for the realm to the propagation configuration file.

EUVF04145I Listening for database propagation on port *number*.

Explanation

The **kpropd** command is ready to receive a database propagation.

User response

Initiate® the database propagation from the primary KDC for the realm.

EUVF04146I Receiving database propagation from *server*.

Explanation

The **kpropd** command is receiving a database propagation.

User response

None

EUVF04147I Network interface status.

Explanation

This message is displayed in response to the Kerberos security server DISPLAY NETWORK command. The remaining lines in this multi-line message display the status of each network interface. A network interface is ACTIVE if the Kerberos security server is listening for requests on that interface. A network interface is INACTIVE if the interface has been stopped and has not been restarted yet. No entry is displayed for network interfaces that have never been active since the security server was started. The Kerberos security server checks for network interface changes based on the value of the SKDC_NETWORK_POLL environment variable, which has a default value of 5 minutes.

User response

None

EUVF04148I No active network interfaces.

Explanation

When issued in response to a DISPLAY NETWORK console command, this message indicates there are no active network interfaces that the KDC is listening for inbound requests. When the KDC is configured to support binding to specific IP addresses, this message indicates that none of the specified IP addresses are active at KDC initialization, or all the specified IP addresses became inactive during KDC operation. See "Configuring KDC bind support" on page 26 for additional information for binding to specific IP address.

User response

When issued in response to a DISPLAY NETWORK console command, no response is necessary. When the KDC is configured to support binding to specific IP addresses, ensure the specified addresses are correct for the system.

EUVF04149R Enter 1 to delete the database or 0 to cancel the request.

Explanation

The **kdb5_ndbm destroy** command has been issued and the user is being prompted to confirm the request to delete the KDC database.

User response

Enter 1 to continue with the delete request or 0 to cancel the delete request.

EUVF04150I Component trace started.

Explanation

The Kerberos component trace has been started. The jobs specified on the TRACE CT command may be already running or may be started after the TRACE CT command is processed. However, any jobs that are already running must have been started after the SKRBKDC started task was started.

User response

None.

EUVF04151I Component trace ended.

Explanation

The Kerberos component trace has ended.

User response

None.

EUVF04152I Component trace started for *jobname*.

Explanation

The Kerberos component trace has started for the indicated job. This message is displayed for each job specified on the TRACE CT command when the application makes its first Kerberos API request after component tracing has been started.

User response

None.

EUVF04153W Component trace buffer overflow.

Explanation

Both of the Kerberos component trace buffers are full and additional trace entries cannot be added until the trace writer has written the current data to the trace dataset. Trace entries will be discarded until the trace writer has emptied one of the trace buffers.

User response

Increase the trace buffer size specified on the TRACE command and restart the component trace.

EUVF04154E Incorrect OPTIONS syntax

Explanation

The OPTIONS parameter syntax on the IPCS CTRACE command is not correct for a Kerberos component trace. Kerberos supports four options: JOB, PID, TID and LVL. The CTRACE OPTIONS parameter is specified as CTRACE COMP(SKRBKDC) OPTIONS((JOB(name), PID(hexid), TID(hexid), LVL(hexdigit))).

User response

Specify a valid OPTIONS parameter.

EUVF04155E Incorrect trace option.

Explanation

An incorrect trace option was specified on the IPCS CTRACE command for a Kerberos component trace. Kerberos supports four options: JOB, PID, TID and LVL. The CTRACE OPTIONS parameter is specified as CTRACE COMP(SKRBKDC) OPTIONS((JOB(name), PID(hexid), TID(hexid), LVL(hexdigit))). The job name must be 1-8 characters. The PID and TID identifiers must be 1-8 hexadecimal digits. The message level must be a single hexadecimal digit.

User response

Specify a valid OPTIONS parameter.

EUVF04156E Duplicate trace option.

Explanation

A Kerberos trace option is specified more than once on the IPCS CTRACE command.

User response

Do not specify the same trace option more than once.

EUVF04157E Incorrect hexadecimal value.

Explanation

The value for the PID, TID and LVL trace options for the IPCS CTRACE command must be hexadecimal values. The PID and TID values may contain 1-8 hexadecimal digits while the LVL value must be a single hexadecimal digit.

User response

Specify a valid hexadecimal value.

EUVF04158I Kerberos KDC services are enabled.

Explanation

The SKRBKDC started task will provide Kerberos Key Distribution Center services. KDC services can be disabled by specifying PARM='-nokdc' instead of PARM='-kdc' when starting the SKRBKDC started task.

User response

None.

EUVF04159I Kerberos KDC services are disabled.

Explanation

The SKRBKDC started task will not provide Kerberos Key Distribution Center services. KDC services can be enabled by specifying PARM='-kdc' instead of PARM='-nokdc' when starting the SKRBKDC started task.

User response

None.

EUVF04160E Unrecognized SKRBKDC start parameter.

Explanation

An unrecognized parameter is specified in the PARMS field of the SKRBKDC started task. The supported parameters are –kdc and –nokdc.

User response

Specify a valid parameter and restart the SKRBKDC started task.

EUVF04163E The KDC requires PKINIT configuration and configuration errors exist.

Explanation

The KDC indicates it requires PKINIT to be used, but not all the PKINIT environment variables are set properly. KDC does not start.

User response

Correct the PKINIT environment variables and restart the KDC.

EUVF04164E Error opening the KDC key ring, key token, or key database.

Explanation

During KDC initialization for PKINIT, the key ring, key token, or key database can not be opened.

User response

Make sure that the value that is specified for the key ring, key token, or key database is correct and the KDC has access to it.

EUVF04165E Error reading the KDC [*key ring* | *key token* | *key database*]: Status status-code – *status-message*

Explanation

During KDC initialization for PKINIT, the certificates or keys can not be retrieved from the key ring, key token, or key database.

User response

Make sure that the key ring, key token, or key database is set up with certificates and keys needed for PKINIT.

EUVF04166I The KDC FIPS level is set to *level*.

Explanation

Indicates the FIPS level the KDC is running. For a description of the FIPS levels for the KDC, see the description for the SKDC_FIPSLEVEL environment variable in "Security server environment variables" on page 32.

User response

None.

EUVF04167E Incorrect value *level* specified for SKDC_FIPSLEVEL.

Explanation

The value indicated in the message is not valid for the SKDC_FIPSLEVEL. The valid values are 0, 1, 2 and 3. The KDC does not start.

User response

Correct the value for SKDC_FIPSLEVEL. Restart KDC once the correction is made.

EUVF04168E The KDC is unable to run in FIPS mode, error 0xnnnn.

Explanation

During KDC initialization, a call to the System SSL function to establish the KDC FIPS level failed with the unexpected error included in the message. As a result, the KDC failed to start.

User response

Refer to the z/OS Cryptographic Services System SSL Programming manual for a description of the error code indicated in the message. If you are unable to determine the cause of the failure, contact your IBM service representative.

EUVF04169E Incompatible encryption type specified for FIPS level *level*.

Explanation

None of the encryption types specified by SKDC_TKT_ENCTYPES are compliant with the level specified by SKDC_FIPSLEVEL environment variable. The KDC does not start. The qualified encryption types for FIPS mode are:

- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha1-96
- des3-cbc-sha1
- aes128-cts-hmac-sha256-128

- aes256-cts-hmac-sha384-192

User response

Change the encryption type or the FIPS level accordingly and restart the KDC.

EUVF04170E Incompatible Diffie-Hellman key size specified for FIPS level *level*.

Explanation

SKDC_PKINIT_REQUIRED is set to 1, but the value specified by SKDC_PKINIT_DH_MIN_BITS is not compliant with the level specified by SKDC_FIPSLEVEL in the envar file. The KDC does not start. When the FIPS level is greater than 0, the required value for SKDC_PKINIT_DH_MIN_BITS is 2048.

User response

Change the value of SKDC_PKINIT_DH_MIN_BITS or the FIPS level accordingly and restart the KDC.

EUVF04171E ICSF is unavailable

Explanation

The KDC determined that ICSF was not available during startup and stopped. ICSF is required to be started and completed initialization prior to starting the KDC, and remain available for the duration of KDC operation.

User response

Start ICSF before starting the KDC.

EUVF04172E Incompatible encryption type *enctype_name* for the *source* master key for FIPS level *fipslevel*.

Explanation

The encryption type of the master key in the indicated source is not FIPS compliant. The encryption types for FIPS mode are:

- aes256-cts-hmac-sha384-192
- aes128-cts-hmac-sha256-128
- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha1-96
- des3-cbc-sha1

User response

Prior to enabling FIPS for an existing Network Authentication Service NDBM database, run the kdb5_ndbm fips_report utility to check for readiness to enable FIPS. If the master key is not FIPS compliant, follow the kdb5_ndbm documentation for the dump and load operation to change the master key encryption type (-mkey_convert). If the kadmin/history principal does not have FIPS compliant keys, update the history key when changing the master key encryption type (-hkey_convert).

EUVF04173I Command Option *option* is ignored

User response

None.

EUVF04174E Unable to read master principal from dump file *dump_file_name*

Explanation

The Kerberos security server is unable to read the master principal from the indicated dump file.

User response

Ensure that the dump file provided in the kdb5_ndbm load command is a valid dump file.

EUVF04175I *no of password entries* Password history entry for *principal_name* could not be checked because of environment constraints.

Explanation

This warning message is issued when the kadmin change_password request is processed when running in FIPS mode and the key entries in the password history are not FIPS compliant. The indicated number of password history entries were not checked for duplication.

User response

For principals that maintain password that have only DES history keys, be aware that password history checks during a *change_password* will be unable to generate a DES key from the new password when FIPS is enabled to compare with the password history keys to detect a password re-use attempt. If this is a required security capability in your environment, configure the environment to use only FIPS compatible keys, but do not enable FIPS until there are no more DES history keys reported from the kdb5_ndbm *fips_report* utility.

EUVF04177I The database master key encryption type is *enctype name*.

Explanation

This message is displayed in response to kdb5_ndbm load command failure when the encryption type of the master key specified using the -k option does not match the encryption type of the master key in the dump file.

User response

The kdb5_ndbm load command failed because the master key encryption type did not match the master key encryption type in the dump file. Specify the indicated master key encryption type in the -k command option and retry the kdb5_ndbm load command.

EUVF04178E Principal contains only Non-FIPS compliant encryption keys.

Explanation

The current principal has only Non-FIPS compliant keys in the NDBM database.

User response

The principal in the NDBM database has only DES and/or DESD keys. Prior to enabling FIPS mode, enable FIPS compliant encryption type as documented in NAS Administration and change the password for the indicated principals. Re-run the utility to verify the issue is resolved.

EUVF04179W Principal contains both FIPS and Non-FIPS compliant encryption keys.

Explanation

The current principal has both FIPS and Non-FIPS compliant keys in the NDBM database

User response

The principal in the NDBM database has a mixture of FIPS and Non-FIPS compliant keys. Prior to enabling FIPS mode, you may enable FIPS compliant encryption type as documented in *z/OS Integrated Security Services Network Authentication Service Administration* and change the password for the indicated principals. Re-run the utility to verify the issue is resolved. This is recommended action but not a mandatory action.

EUVF04180I All principal keys are FIPS compliant.

Explanation

The current principal has all FIPS compliant keys in the NDBM database.

User response

None

EUVF04181E Principal history contains only Non-FIPS compliant encryption keys.

Explanation

The principal history for the current principal has only Non-FIPS compliant keys in the NDBM database.

User response

The principal password history has only DES keys in the NDBM database. The password history checks during change_password will be unable to generate a DES key from the new password when FIPS is enabled to compare with the history keys to detect a password re-use attempt. Prior to enabling FIPS mode, ensure there are no more DES history keys reported from the kdb5_ndbm *fips_report* utility by changing the password for the indicated principals using the change_password kadmin subcommand.

EUVF04182W Principal history contains both FIPS and Non-FIPS compliant encryption keys.

Explanation

The principal history for the current principal has both FIPS and Non-FIPS compliant keys in the NDBM database.

User response

The principal history in the NDBM database has a mixture of FIPS and Non-FIPS compliant keys. Prior to enabling FIPS mode, ensure there are no more DES history keys reported from the kdb5_ndbm *fips_report* utility by changing the password for the indicated principals using the change_password kadmin subcommand.

EUVF04183I All principal history keys are FIPS compliant.

Explanation

The principal history for the current principal has all FIPS compliant keys in the NDBM database.

User response

None

EUVF04183E **Unable to generate the FIPS report: *Status status-code - status-message*.**

Explanation

The database utility command kdb5_ndbm is unable to generate the FIPS report.

User response

Ensure that the master key and history key used are valid. Contact your service representative if the error persists.

Messages for Kerberos commands (numbers EUVF06000 - EUVF06999)

EUVF06001E The *option* option requires value.

Explanation

A command line parameter is specified that requires a value. No value is found.

User response

Specify a value for the command line parameter.

EUVF06002E *option* is not a valid command option.

Explanation

An unrecognized command line parameter is specified.

User response

Specify a valid command line parameter.

EUVF06003E Time delta value *value* is not valid.

Explanation

The time delta specified is not correct. Time deltas are specified as a string of time values with no intervening blanks. Each time value consists of a decimal number followed by **w, d, h, m,** or **s** corresponding to weeks, days, hours, minutes, and seconds. If a number is specified without a letter, it defaults to hours. For example, **1d6h30m** specifies a time delta of 1 day, 6 hours, 30 minutes.

User response

Specify a valid time delta value.

EUVF06004I Usage: kinit [-s] [-c cache_name] [-T armor_ccache] [-k [-t keytab]] [-A] [-f] [-n] [-p] [-R] [-l end] [-r till] [-X attribute[=value]] [principal]

Explanation

This message displays the valid command line options for the **kinit** command.

-s
Use the Kerberos principal associated with the current system identity. No password is used since the system has already verified the identity.

-c
Specify the credentials cache name. The default credentials cache is used if this option is not specified.

-k
Obtain the password from a key table. The user is prompted for the password if this option is not specified.

-t
Specify the name of the key table. The default key table is used if this option is not specified.

-A
Request a ticket that does not contain a client address list.

-f
Request a ticket that can be forwarded.

-p
Request a ticket that a proxy can use.

-R
Renew an existing ticket.

-l
Request a ticket with the specified end time interval.

-r
Request a ticket that can be renewed for the specified time period.

-X
Specify a pre-authentication attribute and value to be passed. This option may be specified multiple times to specify multiple attributes. If -s is specified, this option will not take effect.

For Public Key Cryptography for initial authentication (PKINIT), the attributes are:

- keyring – specify the key ring, key token or key database that contains the end entity certificate and its CA certificate. If a key database is used, its stash file needs to be specified too.
- stash – the stash file contains the password of the key database, ignored if key database is not specified.
- rsa_protocol – specify 'yes' or 'no' to indicate whether to use RSA protocol, if no value is specified, it is defaulted to yes, i.e RSA protocol is used; if this attribute is not specified or a value of no is specified, Diffie-Hellman protocol is used.

-n
Request an anonymous ticket. See "kinit" on page 92.

-T
Specify the credential cache that contains the armor ticket. See "kinit" on page 92.

Delta times are specified as a string of time values with no intervening blanks. Each time value consists of a decimal number followed by **w, d, h, m,** or **s**, corresponding to weeks, days, hours, minutes, and seconds. If a number is specified without a letter, it defaults to hours. For example, **1d6h30m** specifies a time delta of 1 day, 6 hours, 30 minutes.

If no client principal is specified, the default principal for the credentials cache is used.

User response

None

EUVF06005E Unable to parse principal name. Status *status-code* - *status-message*.

Explanation

The **kinit** command is unable to parse the principal name.

User response

Specify a valid principal name on the **kinit** command line.

EUVF06006E No options allowed when renewing or validating ticket.

Explanation

No options may be specified when the -R option is specified for the **kinit** command.

User response

Do not specify any other options when the -R option is specified.

EUVF06007E Unable to obtain name of default credentials cache.

Explanation

The **kinit** command is unable to obtain the default credentials cache name.

User response

Verify that the KRB5CCNAME environment variable, if set, specifies a valid credentials cache name. Contact your service representative if the error persists.

EUVF06008E Unable to resolve credentials cache *name*. Status *status-code* - *status-message*.

Explanation

The **kinit** command is unable to resolve the credentials cache name.

User response

Enter a valid credentials cache name. Contact your service representative if the error persists.

EUVF06009E No initial ticket available.

Explanation

The **kinit** command is unable to renew the initial ticket because no ticket is available in the credentials cache or because the ticket principal does not match the principal specified on the **kinit** command line.

User response

Ensure the credentials cache contains a renewable initial ticket.

EUVF06010E Principal name must be specified.

Explanation

No principal name is specified on the **kinit** command line but the credentials cache does not contain a default principal.

User response

Specify the principal name.

EUVF06011E Unable to retrieve principal from credentials cache *name*. Status *status-code* - *status-message*.

Explanation

The **kinit** command is unable to get the default principal name from the credentials cache.

User response

Verify that the credentials cache is not modified. Contact your service representative if the error persists.

EUVF06012E Unable to retrieve ticket from credentials cache name. Status *status-code - status-message*.

Explanation

The **kinit** command is unable to retrieve the ticket-granting ticket from the credentials cache. The most likely cause is that the ticket-granting ticket has expired.

User response

Use the **kinit** command to obtain a new initial ticket.

EUVF06013E Initial ticket is not renewable.

Explanation

The **kinit** command is invoked with the -R option but the initial ticket in the credentials cache is not renewable.

User response

Use the **kinit** command with the **-r** option to obtain a renewable initial ticket.

EUVF06014E Unable to obtain initial credentials. Status *status-code - status-message*.

Explanation

The **kinit** command is unable to obtain initial credentials from the Kerberos security server.

User response

Ensure the security server is operational and the correct password is entered. Contact your service representative if the error persists.

EUVF06015E Unable to resolve key table *name*. Status *status-code - status-message*.

Explanation

The **kinit** command is unable to resolve the key table name.

User response

Ensure the key table exists and can be accessed. Contact your service representative if the error persists.

EUVF06016E Password is not correct for *name*.

Explanation

The supplied password is not correct.

User response

Provide the correct password for the principal.

EUVF06018E Unable to read password. Status *status-code - status-message*.

Explanation

The **kinit** command is unable to read the password.

User response

Contact your service representative if the error persists.

EUVF06019E Unable to store initial credentials in credentials cache *name*. Status *status-code - status-message*.

Explanation

The **kinit** command is unable to store the new credentials in the credentials cache.

User response

Ensure the credentials cache is available to the user. Contact your service representative if the error persists.

EUVF06020I Usage: klist [[-c] [-f] [-e] [-s] [-a]] [-k [-e] [-t] [-K]] [name]

Explanation

This message displays the valid command line options for the **klist** command.

-c
List the contents of a credentials cache. This option is mutually exclusive with the **-k** option. This option is the default if neither **-c** nor **-k** is specified.

-k
List the contents of a key table. This option is mutually exclusive with the **-c** option.

-a
Display expired tickets.

-e
Display ticket encryption types.

-f
Display ticket flag values.

-s
Set exit status based on valid TGT existence.

-t
Show key table timestamps.

-K
Show key table keys.

If no name is specified, the default credentials cache or key table is used.

User response

None

EUVF06021E ***option-1* and *option-2* may not be specified together**

Explanation

Mutually exclusive options are specified.

User response

Specify just one of the options.

EUVF06022E **No default credentials cache found.**

Explanation

No credentials cache name is specified on the **klist** command line and a default credentials cache was not found.

User response

Use the **kinit** command to create a default credentials cache or specify the name of the credentials cache on the **klist** command line.

EUVF06023E **Unable to resolve credentials cache *name*. Status *status-code* - *status-message*.**

Explanation

The **klist** command is unable to resolve the credentials cache name.

User response

Enter a valid credentials cache name. Contact your service representative if the error persists.

EUVF06024E **Unable to retrieve principal from credentials cache *name*. Status *status-code* - *status-message*.**

Explanation

The **klist** command is unable to get the default principal name from the credentials cache.

User response

Verify that the credentials cache is not modified. Contact your service representative if the error persists.

EUVF06025E **Unable to retrieve ticket from credentials cache *name*. Status *status-code* - *status-message*.**

Explanation

The **klist** command is unable to retrieve a ticket from the credentials cache.

User response

Use the **kinit** command to create a new credentials cache.

EUVF06026E **Unable to decode ticket. Status *status-code* - *status-message***

Explanation

The **klist** command is unable to decode a ticket retrieved from the credentials cache.

User response

Verify that the credentials cache is not modified. Contact your service representative if the error persists.

EUVF06027E **No default key table found.**

Explanation

No key table name is entered on the **klist** command line and a default key table is not found.

User response

Specify the key table name.

EUVF06028E **Unable to resolve key table *name*. Status *status-code* - *status-message*.**

Explanation

The **klist** command is unable to resolve the key table name.

User response

Enter a valid key table name. Contact your service representative if the error persists.

EUVF06029E **Unable to read entry key table *name*. Status *status-code* - *status-message*.**

Explanation

The **klist** command is unable to read an entry from the key table.

User response

Verify that the key table is not modified. Contact your service representative if the error persists.

EUVF06030I **Usage: kdestroy [-c cache_name] [-e exp_delta]**

Explanation

This message displays the valid command line options for the **kdestroy** command.

-c
Specify the credentials cache name. This option is mutually exclusive with the -e option. The default credentials cache is used if this option is not specified.

-e
Specify the credentials expiration time delta value. This option is mutually exclusive with the -c option. The expiration time is computed by subtracting this delta from the current time. A credentials cache is deleted if all of the credentials in the cache have an expiration time earlier than the computed expiration time.

User response

None

EUVF06031E **No default credentials cache found.**

Explanation

No credentials cache name is specified on the **kdestroy** command line and a default credentials cache was not found.

User response

Use the **kinit** command to create a default credentials cache or specify the name of the credentials cache on the **kdestroy** command line.

EUVF06032E **Unable to resolve credentials cache *name*. Status *status-code* - *status-message*.**

Explanation

The **kdestroy** command is unable to resolve the credentials cache name.

User response

Enter a valid credentials cache name. Contact your service representative if the error persists.

EUVF06033E **Unable to destroy credentials cache *name*. Status *status-code* - *status-message*.**

Explanation

The **kdestroy** command is unable to delete the credentials cache.

User response

Verify that the credentials cache exists and is accessible. Contact your service representative if the error persists.

EUVF06034I **Credentials cache *name* destroyed.**

Explanation

The **kdestroy** command has successfully deleted the credentials cache.

User response

None

EUVF06035E **Unable to read credentials cache directory. *error-message***

Explanation

The **kdestroy** command is unable to read the credentials cache directory.

User response

Ensure the credentials cache directory exists and is accessible. Contact your service representative if the error persists.

EUVF06036E **Unable to retrieve ticket from credentials cache *name*. Status *status-code* - *status-message*.**

Explanation

The **kdestroy** command is unable to retrieve a ticket from the credentials cache.

User response

Use the **kinit** command to create a new credentials cache.

EUVF06037E **The name function detects an error. *error-text*.**

Explanation

A system function detected an error. Refer to the documentation for the failing system function to obtain more information about the cause of the failure.

User response

Contact your service representative if the error persists.

EUVF06039E ***value* is not a positive whole number.**

Explanation

A value that is not a positive integer was entered.

User response

Specify a positive integer.

EUVF06041E **You must specify add, delete, list, check or merge.**

Explanation

No operation is specified on the **keytab** command line.

User response

You must specify add, delete, list, check or merge.

EUVF06042E **No default key table found.**

Explanation

No key table name is entered on the **keytab** command line and a default key table was not found.

User response

Specify the key table name.

EUVF06043E **Unable to resolve key table *name*. Status *status-code* - *status-message*.**

Explanation

The **keytab** command is unable to resolve the key table name.

User response

Enter a valid key table name. Contact your service representative if the error persists.

EUVF06044E **Unable to read entry key table *name*. Status *status-code* - *status-message*.**

Explanation

The **keytab** command is unable to read an entry from the key table.

User response

Verify that the key table was not modified. Contact your service representative if the error persists.

EUVF06045E **You must specify the principal name.**

Explanation

The principal name is not specified for a key table add or delete operation.

User response

Specify the principal name.

EUVF06046E **Unable to parse principal name. Status *status-code* - *status-message*.**

Explanation

The **keytab** command was unable to parse the principal name.

User response

Specify a valid principal name on the **keytab** command line.

EUVF06047E **The option *option* is not valid for *operation* request.**

Explanation

An option was specified on the **keytab** command line that is not valid for the requested operation.

User response

Specify options that are supported by the requested operation.

EUVF06048R Enter password:

Explanation

The **keytab** command needs the user password to complete a request.

User response

Enter your password.

EUVF06049R Re-enter password:

Explanation

The **keytab** command needs the user password to complete a request.

User response

Enter your password.

EUVF06050E Unable to read password. Status *status-code* - *status-message*.

Explanation

The **keytab** command is unable to obtain the password from the user.

User response

Contact your service representative if the error persists.

EUVF06051E Unable to add entry to key table *name*.

Explanation

The **keytab** command is unable to add an entry to the key table.

User response

Ensure that the key table is accessible. Contact your service representative if the error persists.

EUVF06052E Unable to remove entry from key table *name*.

Explanation

The **keytab** command was unable to remove an entry from the key table.

User response

Ensure that the key table is accessible. Contact your service representative if the error persists.

EUVF06053E Key version *version* not found for *principal* in keytab *keytab*.

Explanation

The **keytab** command was unable to operate on the requested key version because the key table entry does not exist.

User response

None

EUVF06054E No entries found for *principal* in keytab *keytab*.

Explanation

The **keytab** command was not able to find any key table entries for the specified principal.

User response

None

EUVF06055I Usage: ksetup [-h host] [-n name] [-p password] [-e]

Explanation

This message displays the valid command line options for the **ksetup** command.

-h
Specify the host for the LDAP server. The host is specified as *host-name:port-number*. The default LDAP port of 389 is used if the port number is omitted. The LDAP server specification in the Kerberos configuration file is used if this option is omitted.

-n
Specify the distinguished name to use when binding to the LDAP server. The LDAP_BINDDN environment variables used if this option is omitted.

-p
Specify the password to use when binding to the LDAP server. The LDAP_BINDPW environment variable is used if this option is omitted.

-e
Echo each command to **stdout.**

User response

None

EUVF06056E ***command*** **is not a valid subcommand.**

Explanation

The indicated subcommand is not valid for the **ksetup** command.

User response

Enter a valid subcommand.

EUVF06057I **Valid subcommands are addhost, addkdc, addpwd, addadmin, delhost, delkdc, delpwd, deladmin, listhost, listkdc, listpwd, listadmin, exit.**

Explanation

This message lists the valid subcommands for the **ksetup** command.

User response

None

EUVF06058E **Unable to initialize LDAP client.** ***error-text*****.**

Explanation

The **ksetup** command was unable to initialize the LDAP client runtime. The error text provides additional information on the cause of the failure.

User response

Contact your service representative.

EUVF06059E **Unable to bind to LDAP server.** ***error-text*****.**

Explanation

The **ksetup** command was unable to bind to the LDAP server. The error text provides additional information on the cause of the failure.

User response

Ensure that the LDAP server is operational and that the bind name and password are correct. Contact your service representative if the error persists.

EUVF06060E **Realm name must be specified.**

Explanation

The realm name must be specified on the **ksetup** subcommand.

User response

Specify a realm name.

EUVF06061E **Host name must be specified.**

Explanation

The host name must be specified on the **ksetup** subcommand.

User response

Specify a host name.

EUVF06062E **Too many positional parameters.**

Explanation

Too many positional parameters are specified.

User response

Specify a valid subcommand.

EUVF06063E **Host** ***name*** **already exists.**

Explanation

The host cannot be added to the LDAP directory because it already exists.

User response

None

EUVF06064E **Root domain** ***name*** **is not defined.**

Explanation

The root domain is not defined in the LDAP directory.

User response

Add the root domain to the LDAP directory by adding the appropriate naming suffix entry to the LDAP server configuration file.

EUVF06065E **Realm name** ***name*** **is not valid.**

Explanation

A realm name consists of one or more domain components separated by periods.

User response

Enter a valid realm name.

EUVF06066E The *name* function detects an error. DN: *distinguished-name error-text*

Explanation

An LDAP function detected an error for the specified distinguished name.

User response

Ensure that the LDAP server is operational. Contact your service representative if the error persists.

EUVF06067E Insufficient storage available.

Explanation

There is not enough storage available to process the request.

User response

Increase the amount of storage available to the command.

EUVF06068E Host name *name* is not valid.

Explanation

A host name may not contain a colon.

User response

Enter a valid host name.

EUVF06069E Port number *value* is not valid.

Explanation

The port value is not a valid number.

User response

Enter a valid number.

EUVF06070E Host *name* is not defined.

Explanation

The **ksetup** command was unable to delete the indicated host because it is not defined in the LDAP directory.

User response

None.

EUVF06071E No KDC defined for host *name*.

Explanation

The **ksetup** command was unable to delete the KDC definition because no KDC is defined for the indicated host.

User response

None.

EUVF06072E Unable to obtain default realm name. Status *status-code* - *status-message*.

Explanation

The **ksetup** command is unable to obtain the default realm name.

User response

Ensure the default realm name is set in the Kerberos configuration file. Contact your service representative if the error persists.

EUVF06073E Principal name not allowed with -s option.

Explanation

The principal name may not be specified on the **kinit** command when the **-s** option is specified.

User response

Do not specify a principal name when using the system identity.

EUVF06074E Principal *name* is not valid. Status *status-code* - *status-message*.

Explanation

The principal name specified on the **kvno** command line is not valid.

User response

Enter a valid principal name.

EUVF06075E Network credentials are not available. Status *status-code* - *status-message*.

Explanation

No default network credentials are available.

User response

Use the **kinit** command to create a default credentials cache and then retry the **kvno** command.

EUVF06076E **Unable to obtain temporary credentials cache. Status *status-code - status-message*.**

Explanation

The **kvno** command is unable to create a temporary credentials cache.

User response

Contact your service representative if the error persists.

EUVF06077E **Unable to obtain service ticket for principal *principal*. Status *status-code - status-message*.**

Explanation

An attempt to obtain a service ticket for the service principal has failed. The Status contains a more specific reason for the failure.

User response

Verify that the Kerberos security server is started and that the principal is defined in the KDC and is enabled for service tickets, then retry the command. Contact your service representative if the error persists.

EUVF06078E **Principal *name* is not valid. Status *status-code - status-message*.**

Explanation

The principal name specified on the **kpasswd** command line is not valid.

User response

Enter a valid principal name.

EUVF06079E **Unable to read default credentials cache *name*. Status *status-code - status-message*.**

Explanation

The **kpasswd** command is unable to obtain the principal name from the default credentials cache.

User response

Create a new default credentials cache using the **kinit** command and then retry the **kpasswd** command.

EUVF06080E **No default credentials cache.**

Explanation

There is no default credentials cache for the current user.

User response

Either create a default credentials cache using the **kinit** command or specify the principal name on the **kpasswd** command line.

EUVF06081E **Unable to parse *name*. Status *status-code - status-message*.**

Explanation

The **kpasswd** command is unable to parse the principal name.

User response

Enter a valid principal name and retry the command. Contact your support representative if the error persists.

EUVF06082E **Unable to map user *name* to a Kerberos principal. SAF error *code*, Return code *code*, Reason code *code*.**

Explanation

The **kpasswd** command is unable to map the user ID to a Kerberos principal. Refer to the description of the IRRSIM00 function in the *z/OS Security Server RACF Callable Services* document for more information on the error codes.

User response

Verify that the user ID is defined in the security database with an associated Kerberos principal.

EUVF06083I **Changing password for *principal*.**

Explanation

The **kpasswd** command is changing the password for the indicated principal.

User response

None

EUVF06084R Enter current password:

Explanation

The **kpasswd** command needs the current password for the principal.

User response

Enter the current password.

EUVF06085R Enter new password:

Explanation

The **kpasswd** command needs the new password for the principal.

User response

Enter the new password.

EUVF06086R Re-enter new password:

Explanation

The **kpasswd** command compares both new password values to check for typing errors.

User response

Enter the new password again.

EUVF06087E Unable to read password. Status *status-code - status-message*.

Explanation

The **kpasswd** command is unable to obtain a password from the user.

User response

Contact your service representative.

EUVF06088E Unable to obtain initial ticket. Status *status-code - status-message*.

Explanation

The **kpasswd** command is unable to obtain an initial ticket to the **kadmin/changepw** service.

User response

Verify that the **kadmin/changepw** principal is defined in the security database. Contact your service representative if the error persists.

EUVF06089E Password is not correct for *principal*.

Explanation

The entered password is not correct.

User response

Enter the correct password for the principal. Contact your service representative if the error persists.

EUVF06090E Unable to issue password change request. Status *status-code - status-message*.

Explanation

The **kpasswd** command was unable to send the password change request to the password server.

User response

Verify that the password server is defined properly and is operational. Contact your service representative if the error persists.

EUVF06091E Password change request failed. Error code *error-code - error-message*.

Explanation

The password change request was rejected by the password server.

User response

Verify that the new password is valid for the user and that the user is authorized to change his password. Contact your service representative if the error persists.

EUVF06092E Password change request failed. Error code *error-code - error-message*. Server status:*status-message*.

Explanation

The password change request was rejected by the password server.

User response

Verify that the new password is valid for the user and that the user is authorized to change his password. Contact your service representative if the error persists.

EUVF06093I Password changed.

Explanation

The password has been changed.

User response

None

EUVF06094I Password change canceled.

Explanation

The password change request was canceled because no password was entered in response to the prompt.

User response

None

EUVF06095E No password server defined for host *name*.

Explanation

The **ksetup** command is unable to delete a password server definition because no password server is associated with the indicated host.

User response

None

EUVF06096I Key table add canceled.

Explanation

The key table add request was canceled because no password was entered in response to the prompt.

User response

None

EUVF06097E No administration server defined for host *name*.

Explanation

The **ksetup** command was unable to delete an administration server definition because no administration server is associated with the indicated host.

User response

None

EUVF06098I Usage: kpasswd [-A] principal

Explanation

This message displays the valid command line options for the **kpasswd** command.

-A
Specifies that an address list will not be included in the initial ticket used by the **kpasswd** command.

principal
Specifies the principal whose password is to be changed.

User response

None

EUVF06099I Usage: kadmin [-r realm] [-p principal] [-k keytab] [-w password] [-A] [-e]

Explanation

This message displays the valid command line options for the **kadmin** command.

-r realm
Specifies the Kerberos administration realm. The local realm is used if this option is not specified.

-p principal
Specifies the administrator principal. The string **/admin** is appended to the principal obtained from the credentials cache if this option is not specified.

-k keytab
Specifies the key table containing the password for the administrator principal. The user is prompted to enter the password if neither the **-k** nor the **-w** option is specified.

-w password
Specifies the password for the administrator principal. The user is prompted to enter the password if neither the **-k** nor the **-w** option is specified.

-A
Specifies that an address list will not be included in the initial ticket used by the kadmin command.

-e
Echo each command to **stdout**.

User response

None

EUVF06100E *subcommand* is not a valid subcommand.

Explanation

The specified subcommand is not valid for the **kadmin** command.

User response

Enter a valid subcommand.

EUVF06101I **Valid subcommands:**
list_principals (listprincs)
get_principal (getprinc)
add_principal (addprinc)
delete_principal (delprinc)
modify_principal (modprinc)
rename_principal (renprinc)
change_password (cpw)
list_policies (listpols)
get_policy (getpol)
add_policy (addpol)
modify_policy (modpol)
delete_policy (delpol)
get_privs (getprivs)
add_key (ktadd)exit (quit)

Explanation

This message lists the valid subcommands for the **kadmin** command.

User response

None

EUVF06102E **Unable to obtain principal from default credentials cache. *Status-code - Error-text.***

Explanation

The **kadmin** command was unable to obtain the principal from the default credentials cache.

User response

Verify that the credentials cache can be accessed and has not been modified.

EUVF06103E **Unable to determine the administration principal.**

Explanation

The **kadmin** is unable to determine the administration principal.

User response

Either specify the **-p** command line option for the **kadmin** command or use the **kinit** command to set up a default credentials cache.

EUVF06104E **Unable to initialize connection with administration server.**

Explanation

The **kadmin** command is unable to establish a connection with the Kerberos administration server.

User response

Verify that the administration server is running and then retry the request.

EUVF06105E **Unable to perform *request-type* administration request.**

Explanation

The **kadmin** command is unable to perform the requested administration function.

User response

Verify that the administration server is running and then retry the request. Contact your service representative if the error persists.

EUVF06106E **Too many positional parameters.**

Explanation

Too many positional parameters are specified.

User response

Specify a valid subcommand.

EUVF06107I **list_principals [expression]**

Explanation

This message displays the syntax for the **list_principals** subcommand.

User response

None

EUVF06108E **Unable to obtain default realm. *Status-code - Error-text.***

Explanation

The **kadmin** command is unable to get the default realm from the Kerberos configuration file.

User response

Verify that the default realm is defined in the Kerberos configuration file.

EUVF06109E No name specified.

Explanation

No object name is specified on a **kadmin** subcommand.

User response

Specify an object name.

EUVF06110I get_principal name

Explanation

This message displays the syntax for the **get_principal** subcommand.

User response

None

EUVF06111E Unable to parse principal name *name*.

Explanation

The **kadmin** command is unable to parse the indicated principal name.

User response

Specify a valid principal name.

EUVF06112I add_principal [options] [attributes] name

Explanation

This message displays the syntax for the **add_principal** subcommand.

User response

None

EUVF06113E The *name* option requires a value.

Explanation

The indicated **kadmin** subcommand option requires a value but no value was specified.

User response

Specify a value for the option.

EUVF06114E The date specified for the *option* option is not valid.

Explanation

The date is not valid. Refer to the *z/OS Integrated Security Services Network Authentication Service Programming* for a description of the valid date formats. This error can also occur if the specified date is in the past.

User response

Specify a valid date.

EUVF06115E Key version *version* is not valid.

Explanation

The key version is not valid. The key version is an unsigned number between 1 and 2147483647.

User response

Specify a valid key version.

EUVF06116R Enter password:

Explanation

The **kadmin** command is waiting for the user to enter the password.

User response

Enter the password for the principal.

EUVF06117R Re-enter password:

Explanation

The **kadmin** command is waiting for the user to re-enter the password. The re-entered password must match the password that was entered previously.

User response

Enter the password for the principal.

EUVF06118E Password is too long.

Explanation

The password is longer than 128 characters.

User response

Enter a shorter password.

EUVF06119E Unable to read password. Status *status-code - status-message*.

Explanation

The **kadmin** command is unable to read the password.

User response

Contact your service representative if the error persists.

EUVF06120I list_policies [expression]

Explanation

This message displays the syntax for the **list_policies** subcommand.

User response

None

EUVF06121I get_policy name

Explanation

This message displays the syntax for the **get_policy** subcommand.

User response

None

EUVF06122I Principal *name* added.

Explanation

The indicated principal has been added to the Kerberos database.

User response

None

EUVF06123I delete_principal name

Explanation

This message displays the syntax for the **delete_principal** subcommand.

User response

None

EUVF06124I Principal *name* deleted.

Explanation

The indicated principal has been deleted from the Kerberos database.

User response

None

EUVF06125I modify_principal [options] [attributes] name

Explanation

This message displays the syntax for the **modify_principal** subcommand.

User response

None

EUVF06126I Principal *name* modified.

Explanation

The indicated principal has been modified in the Kerberos database.

User response

None

EUVF06127I rename_principal oldname newname

Explanation

This message displays the syntax for the **rename_principal** subcommand.

User response

None

EUVF06128I Principal *oldname* renamed to *newname*.

Explanation

The indicated principal has been renamed in the Kerberos database.

User response

None

EUVF06129E Subcommand is too long.

Explanation

The maximum length of a **kadmin** subcommand is 1023 characters.

User response

Enter a shorter subcommand string.

EUVF06130I change_password [-randkey | -pw password] [-keepold] [-e keytypes] name

Explanation

This message displays the syntax for the **change_password** subcommand.

User response

None

EUVF06131I Random keys generated for *name*.

Explanation

Random keys have been generated for the indicated principal.

User response

None

EUVF06132I Administration request cancelled.

Explanation

The user canceled the current **kadmin** request by entering a zero-length password when prompted.

User response

None

EUVF06133I Password changed for *name*

Explanation

The password has been changed for the indicated principal.

User response

None

EUVF06134E *value* is not a valid numeric value.

Explanation

A positive number is required.

User response

Enter a valid number.

EUVF06135I add_policy [options] name

Explanation

This message displays the syntax for the **add_policy** subcommand.

User response

None

EUVF06136I Policy *name* added.

Explanation

The indicated policy has been added to the Kerberos database.

User response

None

EUVF06137I modify_policy [options] name

Explanation

This message displays the syntax for the **modify_policy** subcommand.

User response

None

EUVF06138I Policy *name* modified.

Explanation

The indicated policy has been modified in the Kerberos database.

User response

None

EUVF06139I delete_policy name

Explanation

This message displays the syntax for the **delete_policy** subcommand.

User response

None

EUVF06140I **Policy *name* deleted.**

Explanation

The indicated policy has been deleted from the Kerberos database.

User response

None

EUVF06141I **help [subcommand]**

Explanation

This message displays the syntax for the help subcommand.

User response

None

EUVF06142I **exit**

Explanation

This message displays the syntax for the exit subcommand.

User response

None

EUVF06143I **add_key [[-keytab | -k] keytab_name] [-keepold] [-e keytypes] principal_name**

Explanation

This message displays the syntax for the **add_key** subcommand.

User response

None

EUVF06144I **Keys generated for *principal* and added to *keytab*.**

Explanation

Random keys have been generated for the indicated principal.

User response

None

EUVF06145E **Unable to open key table *name*. Status *status-code - status-message*.**

Explanation

The **kadmin** command is unable to open the indicated key table.

User response

Verify that the key table exists and can be accessed. Contact your service representative if the error persists.

EUVF06146E **Unable to add entry to key table *name*. Status *status-code - status-message***

Explanation

The **kadmin** command is unable to add an entry to the indicated key table.

User response

Verify that the key table can be accessed. Contact your service representative if the error persists.

EUVF06147I **Authenticating as *name*.**

Explanation

The **kadmin** command is obtaining an initial ticket for the indicated principal. The administration privileges associated with this principal are used for subsequent administration requests.

User response

None

EUVF06148I **get_privs**

Explanation

This message displays the syntax for the **get_privs** subcommand.

User response

None

EUVF06149E **Encryption type *name* is not valid.**

Explanation

An unrecognized encryption type was specified.

User response

Specify a valid encryption type.

EUVF06150E Salt type *name* is not valid.

Explanation

An unrecognized salt type was specified.

User response

Specify a valid salt type.

EUVF06151E Connection broken with the administration server.

Explanation

The connection with the administration server has been broken. This indicates either a network problem or a server failure.

User response

Verify the network connectivity with the administration server. Contact your service representative if the error persists.

EUVF06152E principal *principal* version *version* exists in keytab keytab.

Explanation

When adding a principal, or merging a keytab, the principal already existed in the keytab, and could not be replaced. Duplicate entries in the target keytab are not replaced.

User response

Specify the `replace` option if you want to replace keytab entries.

EUVF06153E keytab *keytab* is empty.

Explanation

The keytab does not contain any principals that can be operated on.

User response

Add the required principals to the keytab.

EUVF06154I Encryption type *encryption_type* is not supported.

Explanation

During a keytab operation at least one entry has been encountered using this encryption type, which is not supported by this release of z/OS Network Authentication Service. Entries that have unsupported encryption types are not merged. Other entries for the same principal that contain supported encryption types may still be merged.

User response

None.

EUVF06155E Usage: keytab add principal [-r] [-p *password*] [-v *version*] [-k *keytab*]

Explanation

This message displays information for the keytab command. For more information see “keytab” on page 90.

User response

None.

EUVF06156E Usage: keytab delete principal [-v *version*] [-k *keytab*]

Explanation

This message displays information for the keytab command. For more information see “keytab” on page 90.

User response

None.

EUVF06157E Usage: keytab list [principal [-v *version*]] [-k *keytab*]

Explanation

This message displays information for the keytab command. For more information see “keytab” on page 90.

User response

None.

EUVF06158E Usage: keytab merge in_keytab [principal [-v *version*]] [-r] [-k *keytab*]

Explanation

This message displays information for the keytab command. For more information see “keytab” on page 90.

User response

None.

EUVF06159E Cannot merge a keytab into itself.

Explanation

The source and the target of the keytab merge command are identical.

User response

Specify different names for the source and the target of the keytab merge.

EUVF06160E No entry found for principal *principal*, version *version*, encryption *encryption* in keytab *keytab*.

Explanation

Keytab entries exist for this principal but none that match the version number and the encryption type.

User response

Ensure that the keytab entry version number matches the version number in the KDC, and that an entry exists for each supported encryption type that is used in the current environment.

EUVF06161E Failed decrypting ticket for principal *principal*, version *version*, encryption *encryption* in keytab *keytab* Status *status*.

Explanation

Keytab entries exist for this principal, version number and encryption type which matches the KDC entry, but the decryption of the ticket using this entry has failed. The status code contains more information, but the most common reason is a mismatching key due to an incorrect password.

User response

Ensure that the password used matches exactly what was specified in the KDC entry for the same principal and version number was created. The password is case sensitive, and may need to be entered in upper case, if a RACF database that is not mixed case was used to enter the key in the KDC. Spaces in the password may need to be escaped, or the entire password enclosed in quotes depending on your shell.

EUVF06162E Usage: keytab check [principal] [-k *key* tab]

Explanation

This message displays information for the keytab command. For more information see “keytab” on page 90.

User response

None.

EUVF06163E Unsupported or missing attribute name: [*attribute_name*]

Explanation

This message displays the unsupported attribute name specified with the -X option in the kinit command or in the krb5.conf file. The unsupported or missing name that is specified is displayed. It might be truncated if it is too long. The supported attribute names are:

- keyring (command line) or pkinit_keyring (krb5.conf) – Specify the key ring, key token or key database that contains the end entity certificate and its CA certificate. If a key database is used, its stash file needs to be specified too.
- stash (command line) or pkinit_keyring_stash (krb5.conf)– The stash file contains the password of the key database. It must be specified if a key database is specified, otherwise it is ignored.
- rsa_protocol (command line) or pkinit_rsa_protocol (krb5.conf) – Whether to use RSA protocol, if no value is specified, rsa protocol is used; if this attribute is not specified, DH protocol is used.

User response

Ensure that the attribute name is correct in the command line or in the krb5.conf file.

EUVF06164E The value specified for the [*attribute_name*] attribute is not valid

Explanation

This message displays the invalid value for the attribute specified with the -X option in the kinit command or in the krb5.conf file. The valid values are:

- keyring (command line) or pkinit_keyring (krb5.conf) – In the form of <userid/ringName> or <ringName>, or <*TOKEN*/tokenName> or the full path name of the key database.
- stash (command line) or pkinit_keyring_stash (krb5.conf) – The full path name of the stash file if key database is specified.
- rsa_protocol (command line) – yes or no, or pkinit_rsa_protocol (krb5.conf) – 1 or 0

User response

Ensure that the attribute value is correct in the command line or in the krb5.conf file.

EUVF06165E **Unable to open the key ring, key token, or key database: [*value*].**

Explanation

The kinit command is unable to open the key ring, key token, or key database to retrieve the certificates that are needed for Public Key Cryptography for initial authentication (PKINIT).

User response

Ensure that the key ring, key token or key database exists.

EUVF06166E **Unable to retrieve the certificate for PKINIT. Status *status-code – status-message*.**

Explanation

The kinit command cannot find the certificates from the specified key ring, key token, or key database that are needed for Public Key Cryptography for initial authentication (PKINIT).

User response

Ensure that the key ring, key token or key database contains the valid certificates.

EUVF06167W **One or more values specified in the configuration file are not valid, default values used. Status *status-code – status-message*.**

Explanation

In the client's configuration file, for example, krb5.conf, one or more values that are specified for Public Key Cryptography for initial authentication (PKINIT) with keyword pkinit_rsa_protocol, pkinit_dh_min_bits or pkinit_require_revocation_checking are not valid. The value or values that are specified are ignored and the default value or values that are for that keyword or keywords are used for processing. (Note: If 'ldap' is specified for pkinit_require_revocation_checking, but pkinit_ldap_server is not, pkinit_require_revocation uses the default value too.) The default values are:

pkinit_rsa_protocol: 0
pkinit_dh_min_bits: 2048
pkinit_require_revocation: none

User response

Correct the value in the configuration file for the future use.

EUVF06168E **The certificates set up for PKINIT are not valid. Status *status-code – status-message*.**

Explanation

The key ring, key token or key database is not set up correctly for Public Key Cryptography for initial authentication (PKINIT).

User response

Make sure the key ring, key token or key database contains the certificates needed.

EUVF06169E **The certificate used for PKINIT does not meet the requirements. Status *status-code – status-message*.**

Explanation

The kinit command found the certificate for Public Key Cryptography for initial authentication (PKINIT). However, the certificate does not meet the requirements of the KDC. Check the status code for more details.

User response

Ensure the certificate contains the values required by the KDC. Look at the trace file if the status code does not provide enough details.

EUVF06170E **The KDC's certificate used for PKINIT is not acceptable. *status-code – status-message*.**

Explanation

When the client is processing the reply from the KDC, the certificate used by the KDC does not meet the

requirement of the client's local policy. Check the status code for more details.

User response

Contact the KDC. Look at the trace file if the status code does not provide enough details.

EUVF06171E PKINIT must be configured to obtain an anonymous ticket.

Explanation

The kinit command cannot obtain an anonymous ticket because the Kerberos configuration lacks support for PKINIT. To obtain an anonymous ticket, the KDC must support PKINIT, and the client configuration must be able to verify the KDC PKINIT certificate.

User response

Ensure the KDC is configured to support PKINIT and that the client Kerberos configuration is configured to trust the KDC's certificate.

EUVF06172E Incompatible encryption type(s) specified for the requested FIPS level *fipslevel*.

Explanation

None of the encryption types specified in the –e option is compliant with the level specified by fipslevel in the Kerberos configuration file. The encryption types for FIPS mode are:

- aes256-cts-hmac-sha384-192
- aes128-cts-hmac-sha256-128
- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha1-96
- des3-cbc-sha1

User response

Change the encryption type or the FIPS level accordingly. Rerun the application once the correction is made.

EUVF06173E Unable to retrieve encryption types. *Status-code – Status-message*.

Explanation

During command processing, no encryption types were specified in the –e command, the program tried to retrieve encryption types from the Kerberos configuration file and failed.

User response

Look up the Kerberos error code in the error message.

EUVF06174E Incompatible attribute *attribute_name* specified for principal *principal_name* with FIPS level *fipslevel*.

Explanation

During command processing, support_desmd5 attribute specified is not compatible with the level specified by the SKDC_FIPSLEVEL in the KDC envar file.

User response

The support_desmd5 attribute is not compatible with FIPS. When running in FIPS mode, this attribute is not allowed to be associated with a principal.

EUVF06175I Encryption type *enctype_name* is not compatible for the requested FIPS level *fipslevel*.

Explanation

During keytab merge command processing, this message is issued if the key to be merged is not FIPS compliant if the caller is running in FIPS mode. The merge operation continues without the Non-FIPS compliance keys.

User response

Ensure that at least one encryption type specified is compatible with the level specified by fipslevel in the Kerberos configuration file. The encryption types for FIPS mode are:

- aes256-cts-hmac-sha384-192
- aes128-cts-hmac-sha256-128
- des3-cbc-sha1
- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha1-96

Chapter 8. Component Trace

This chapter presents component trace information for Network Authentication Service for z/OS.

The SKRBKDC started task provides component trace support for any Kerberos application running on the same system as the SKRBKDC started task. The trace records can be written to a trace external writer or they can be kept in an in-storage trace buffer which is part of the SKRBKDC address space. For more information on writing to a trace buffer, see the Tracing applications topic of *z/OS MVS Programming: Authorized Assembler Services Guide*.

IPCS is used to format and display the trace records from either a trace dataset or an SVC dump of the SKRBKDC address space. Refer to *z/OS MVS Diagnosis: Tools and Service Aids* for more information on setting up and using component trace. Refer to *z/OS MVS System Commands* for more information on the TRACE command. Refer to *z/OS MVS IPCS User's Guide* for more information on using IPCS to view a component trace.

Capturing Component Trace Data

The component trace can be started before the job to be traced is started or while the job is running. The trace will be active for the first instance of the job. For example, if the same job name is used for multiple jobs, only the first job with that name will be traced. Subsequent jobs with the same name will not be traced unless the component trace is stopped and then restarted.

A trace external writer is required if the trace records are to be written to a dataset. A sample started procedure is shipped as EUVF.SEUVFSAM(SKRBWTR). Copy this procedure to SYS1.PROCLIB(SKRBWTR) and modify as necessary to meet your installation requirements. The following MVS operator command will start the trace external writer:

```
TRACE CT,WTRSTART=SKRBWTR
```

A single Kerberos component trace may be active at a time and the trace can include from 1 to 16 separate jobs. The SKRBKDC started task will be traced if no job names are specified. The trace buffer size must be between 64K and 512K and will default to 64K.

The OPTIONS parameter specifies the list of subcomponent trace levels as OPTIONS=(subcomp1.lvl1,subcomp2.lvl2,...). Trace messages for a particular subcomponent will not be logged unless the subcomponent is included in the trace list and the message level is greater than or equal to the specified level. An asterisk (*) may be used to specify all subcomponents. Trace level 1 generates the minimum amount of trace message output, trace level 8 generates the maximum amount of trace message output, and trace level 9 generates data dumps in addition to the trace messages. The subcomponent list consists of a subcomponent name and a trace level separated by a period. Multiple subcomponents may be specified by separating the entries with commas.

The following example will enable trace level 1 for all subcomponents and trace level 8 for the KRB_CCACHE subcomponent.

```
OPTIONS=(*.1,KRB_CCACHE.8)
```

The following command will start a Kerberos component trace for jobs CS390IP and DB1G which includes all nondump trace entries and writes the trace records using the SKRBWTR trace writer:

```
TRACE CT,ON,COMP=SKRBKDC
R n,JOBNAME=(CS390IP,DB1G),OPTIONS=(*.8),WTR=SKRBWTR,END
```

The following commands will stop the Kerberos component trace and close the trace writer dataset:

```
TRACE CT,OFF,COMP=SKRBKDC
TRACE CT,WTRSTOP=SKRBWTR
```

Kerberos does not require a default trace member in SYS1.PARMLIB since the Kerberos component trace is not activated until the operator enters the TRACE command. SYS1.PARMLIB members can be created for frequently used trace commands and the member name can then be specified on the TRACE command to avoid the operator prompt for trace options. Sample entries CTIKDC00 and CTIKDC01 are shipped in EUVF.SEUVFSAM for tracing the SKRBKDC started task.

Displaying the Trace Data

The trace records are displayed using the IPCS CTRACE command. The CTRACE ENTIDLIST parameter specifies the trace entries to be included in the display. The trace entry identifier is the Kerberos subcomponent number. All trace entries will be included if the ENTIDLIST parameter is not specified. The following subcomponent numbers are defined:

Table 13. Subcomponent numbers

Entry ID	Mnemonic	Mnemonic Description
0	KRB_API	Kerberos API entry/exit
1	KRB_GENERAL	General Kerberos messages
2	KRB_CCACHE	Credentials cache messages
3	KRB_RCACHE	Replay cache messages
4	KRB_CRYPTO	Cryptography messages
5	KRB_GSSAPI	GSS-API messages
6	KRB_KEYTAB	Key table messages
7	KRB_LIB	Kerberos library messages
8	KRB_ASN1	ASN.1 messages
9	KRB_OS	Operating system interface messages
10	KRB_KDC	KDC messages
11	KRB_KDB	Kerberos database messages
12	KRB_KUT	Kerberos utility messages
13	KRB_RPC	RPC messages
14	KRB_ADMIN	Kerberos administration messages

The CTRACE OPTIONS parameter specifies additional filtering for the trace records. The JOB(name), PID(hexid), TID(hexid), and LVL(number) options can be specified to filter the trace entries based on job name, process identifier, thread identifier, and message level. All trace entries will be included if the OPTIONS parameter is notspecified.

The JOBNAME parameter on the CTRACE command is used to select the address space in a dump. Since the address space is always the SKRBKDC address space, this parameter cannot be used to filter the trace entries. Instead, you must use the OPTIONS((JOB(name))) parameter to select the component trace entries for a specific job.

The following example shows how to display Kerberos API and General trace messages for job KRBSRV48 thread 6 and exclude data dumps:

```
IPCS CTRACE COMP(SKRBKDC) ENTIDLIST(0,1)
OPTIONS((JOB(KRBSRV48),TID(6),LVL(8))) FULL
```

A range can be specified for the entry identifiers. The following examples show how to display just Kerberos DLL trace records:

```
IPCS CTRACE COMP(SKRBKDC) ENTIDLIST(0:9,13:14) FULL
```

Appendix A. Sample Kerberos configurations

In this topic IBM provides sample configurations for three Kerberos realms:

- The KRB390.IBM.COM realm with the KDC on z/OS using z/OS Kerberos. The DNS domain is **krb390.ibm.com**. The sysplex contains two systems with host names **dcesec4.krb390.ibm.com** and **dcesec7.krb390.ibm.com**.
- The KRB2003.IBM.COM realm with the KDC on the Microsoft Windows 2003 operating system using Windows 2003 Server. The DNS domain is **krb2003.ibm.com** and the domain controller is **sstone1.krb2003.ibm.com**.
- The MITKRB.IBM.COM realm with the KDC on AIX using MIT Kerberos 1.2.1. The DNS domain is **mitkrb.ibm.com** and the KDC is located on the **dcecpt.mitkrb.ibm.com** system.

  The DNS server for this example is located on Windows 2003.

The z/OS commands used in these examples assume that the external security manager is RACF.

KRB390.IBM.COM configuration

For this configuration, do these steps:

1. Network Authentication Service for z/OS supports seven encryption types: 56-bit DES (DES), 56-bit DES with key derivation (DESD), 168-bit triple DES (DES3), 128-bit AES (AES128), 256-bit AES (AES256), 128-bit AES SHA2(AES128SHA2), and 256-bit AES SHA2(AES256SHA2). By default, only AES, DES3, and DES are enabled. The use of these encryption keys can be controlled on an individual user basis through the ENCRYPT option of the KERB keyword on the ALTUSER and RALTER commands.
2. By default, the z/OS KDC uses DES, aes256-cts-hmac-sha1-96, aes128-cts-hmac-sha1-96, and des3-cbc-sha1-kd to encrypt tickets. If you want to enable the use of the additional encryption algorithms when encrypting tickets, add the following line to **/etc/skrb/home/kdc/envar**:

   ```
   SKDC_TKT_ENCTYPES=aes256-cts-hmac-sha384-192,aes128-cts-hmac-sha256-128,
   aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1-kd,des-hmac-sha1,
   des-cbc-crc
   ```

3. Copy the initial configuration file from **/usr/lpp/skrb/examples/krb5.conf** to **/etc/skrb/krb5.conf** and edit the entries accordingly. The following entries are set for the sample shown:
   - the *default_realm* value is set to the local realm
   - the *kdc_default_options value* is set to request forwardable tickets
   - the *use_dns_lookup* value is set to use the DNS name server instead of the [realms] and [domain_realm] sections of the configuration file
   - the *pkinit_keyring* is set to a RACF key ring named KRBRING owned by KRBUSR
   - the *pkinit_require_revocation_checking* is set to none, that is no revocation checking will be performed on the KDC certificates
   - the *pkinit_rsa_protocol* is set to 0, that is the client will use the Diffie-Hellman exchange method to encrypt the reply
   - the *pkinit_dh_min_bits* is set to 2048, that is the initial DH key size generated is 2048

   For completeness, the [realms] and [domain_realm] sections are set but aren't needed since the DNS name server is used to locate Kerberos services (SRV records) and to resolve host names to realm names (TXT records). The entries that start with pkinit are needed for using a certificate for authentication.

   ```
   [libdefaults]
   default_realm = KRB390.IBM.COM
   ```

```
kdc_default_options = 0x40000010
use_dns_lookup = 1
[realms]
KRB390.IBM.COM = {
    kdc = dcesec4.krb390.ibm.com:88
    kdc = dcesec7.krb390.ibm.com:88
    kpasswd_server = dcesec4.krb390.ibm.com:464
    kpasswd_server = dcesec7.krb390.ibm.com:464
    pkinit_kdc_hostname=dcesec4.krb390.ibm.com
    pkinit_kdc_hostname=dcesec7.krb390.ibm.com
    pkinit_keyring = KRBUSR/KRBRING
    pkinit_require_revocation_checking = none
    pkinit_rsa_protocol = 0
    pkinit_dh_min_bits = 2048
}
[domain_realm]
.krb390.ibm.com = KRB390.IBM.COM
```

4. Define the default realm attributes in the RACF database. The password can be any value, but the resource name must be KERBDFLT. The KERBNAME value specifies the name of the local Kerberos realm. For this example, the minimum ticket life is 15 seconds, the default ticket life is 10 hours, and the maximum ticket life is 24 hours. Note that the realm name is converted to uppercase by the RDEFINE command but the password is unchanged.

```
RDEFINE REALM KERBDFLT KERB(KERBNAME(KRB390.IBM.COM)
    PASSWORD(74427532) MINTKTLFE(15) DEFTKTLFE(36000)
    MAXTKTLFE(86400))
```

5. Define the password change service. The user name and password can be any value but the Kerberos principal must be **kadmin/changepw**. Note that the password is converted to uppercase by the ALTUSER command if the MIXEDCASE SETROPTS option is not set, but the principal is unchanged.

```
ADDUSER CHANGEPW DFLTGRP(SYS1) PASSWORD(TEMPPASS)
ALTUSER CHANGEPW PASSWORD(74427533) NOEXPIRED
    KERB(KERBNAME(kadmin/changepw))
```

6. Create Kerberos principals for the samples shipped in **/usr/lpp/skrb/examples/gssapi_test** and **/usr/lpp/skrb/examples/krbmsg_test**. The user name and password can be any value but the Kerberos principal names must match the values coded in the examples. Note that the password is converted to uppercase by the ALTUSER command but the principal is unchanged.

```
ADDUSER KRBSRV4 DFLTGRP(SYS1) PASSWORD(TEMPPASS)
ALTUSER KRBSRV4 PASSWORD(TEST4SRV) NOEXPIRED
    KERB(KERBNAME(test_server/dcesec4.krb390.ibm.com))
ADDUSER KRBSRV7 DFLTGRP(SYS1) PASSWORD(TEMPPASS)
ALTUSER KRBSRV7 PASSWORD(TEST7SRV) NOEXPIRED
    KERB(KERBNAME(test_server/dcesec7.krb390.ibm.com))
ADDUSER KRBDLG4 DFLTGRP(SYS1) PASSWORD(TEMPPASS)
ALTUSER KRBDLG4 PASSWORD(TEST4DLG) NOEXPIRED
    KERB(KERBNAME(test_delegate/dcesec4.krb390.ibm.com))
ADDUSER KRBDLG7 DFLTGRP(SYS1) PASSWORD(TEMPPASS)
ALTUSER KRBDLG7 PASSWORD(TEST7DLG) NOEXPIRED
    KERB(KERBNAME(test_delegate/dcesec7.krb390.ibm.com))
ADDUSER KRBCLNT DFLTGRP(SYS1) PASSWORD(TEMPPASS)
ALTUSER KRBCLNT PASSWORD(TESTPSWD) NOEXPIRED
    KERB(KERBNAME(test_client))
```

7. Because the SKRBKDC started task is running on each system in the sysplex, there is no need to set up key tables for use by applications. Instead, the KRB5_SERVER_KEYTAB environment variable is set to 1.

   With RACF as the external security manager, the IRR.RUSERMAP resource in the FACILITY class must be defined. The **test_server** and **test_delegate** system IDs (for example, KRBSRV4 and KRBDLG4) must have RACF READ access to IRR.RUSERMAP resource to use the KRB5_SERVER_KEYTAB variable set to 1. To define IRR.RUSERMAP and grant READ authority to all system users:

```
REDEFINE FACILITY IRR.RUSERMAP UACC(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

See “Security runtime environment variables” on page 26 for more on the KRB5_SERVER_KEYTAB environment variable.

8. Create SRV records for the **_kerberos** and **_kpasswd** services using the UDP and TCP protocols. The example that follows uses the Microsoft® Windows® 2003 DNS management console to create the SRV entries for the **krb390.ibm.com** domain. The SKRBKDC started task is running on both the **dcesec4** and **dcesec7** systems, so there are **_kerberos** and **_kpasswd** entries for both systems. The Kerberos runtime randomly selects entries with the same priority when attempting to contact a service provider, so this example uses the same priority for all entries to provide rudimentary load balancing.

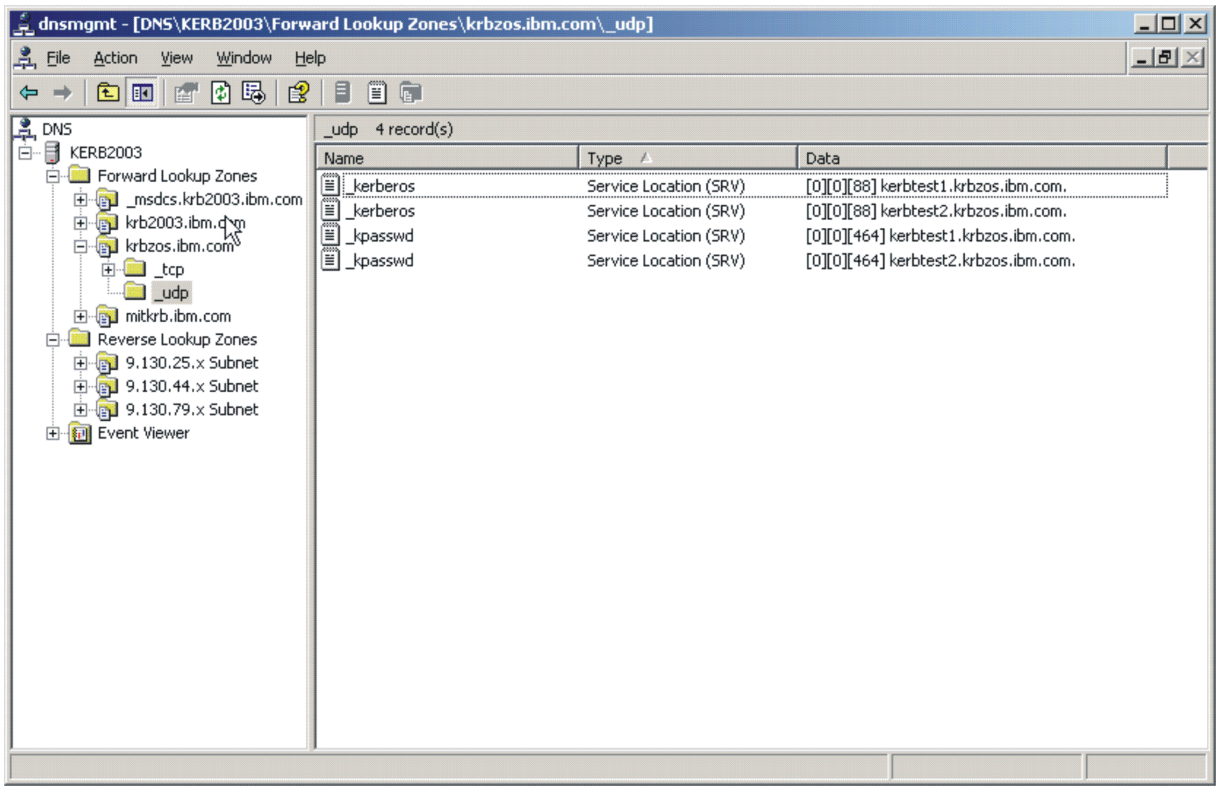

Figure 4. KRB390.IBM.COM configuration - creating SRV entries

9. Create a TXT record to map host names in the **krb390.ibm.com** DNS domain to the KRB390.IBM.COM Kerberos realm. This example uses the Windows 2003 DNS management console to create the TXT record.

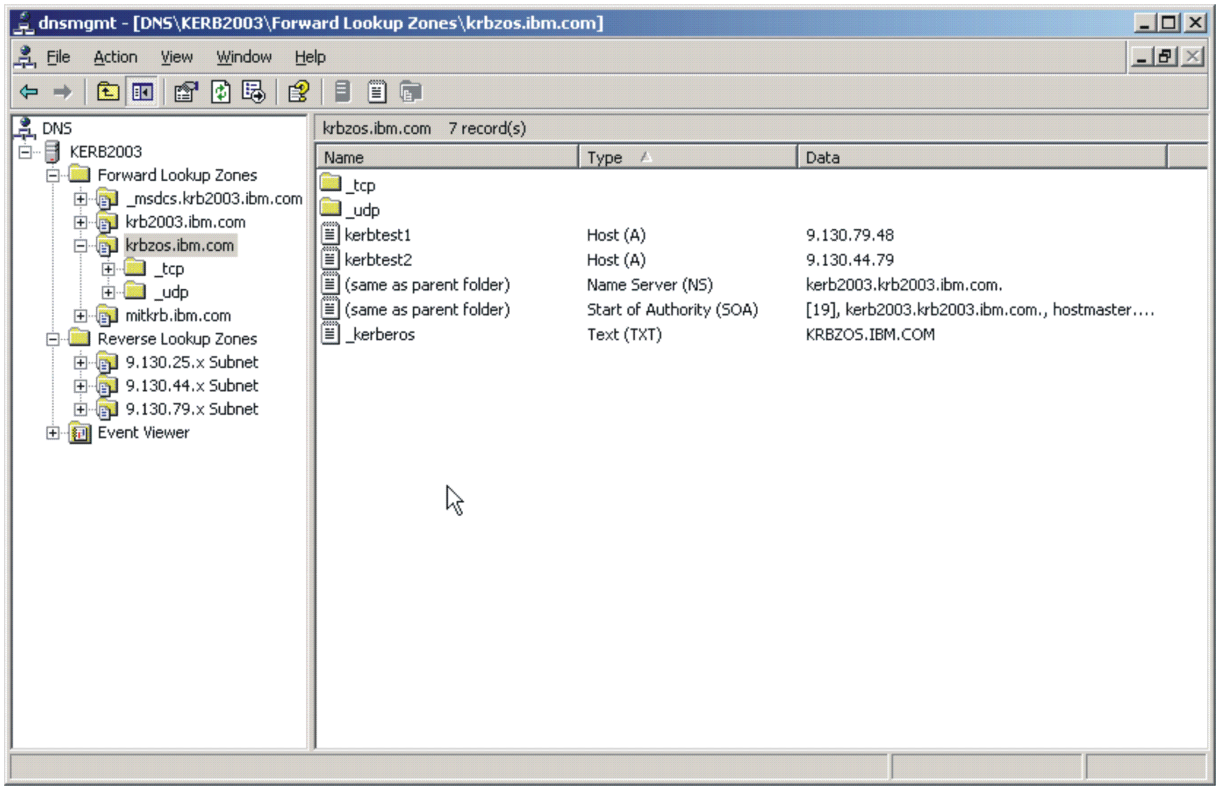

Figure 5. KRB390.IBM.COM configuration - creating a TXT record

10. Additional steps to set up PKINIT:

    a. Generate a self-signed certificate to represent the local certificate authority. This certificate is used as the certificate authority certificate.

```
RACDCERT CERTAUTH
GENCERT
SUBJECTSDN(OU('Local Certificate Authority')
O('Network group')
C('US'))
KEYUSAGE(CERTSIGN)
WITHLABEL('Network Local CA')
```

    b. Generate the KDC certificate

```
RACDCERT ID(SKRBKDC)
GENCERT
SUBJECTSDN(CN('z/OS KDC')
OU('Test')
O('Network A')
C('US'))
ALTNAME(DOMAIN('dcesec4.krb390.ibm.com'))
KEYUSAGE(HANDSHAKE)
WITHLABEL('zOS KDC')
SIGNWITH(CERTAUTH
LABEL('Network Local CA'))
```

    c. Create the KDC keyring

```
RACDCERT ID(SKRBKDC) ADDRING(KDCRING)
```

    d. Connect the KDC certificate to the key ring and mark it as the default certificate.

```
RACDCERT ID(SKRBKDC)
CONNECT(LABEL('zOS KDC')
RING(KDCRING)
DEFAULT)
```

    e. Connect the local certificate authority certificate to the key ring as well.

```
RACDCERT ID(SKRBKDC)
CONNECT(CERTAUTH LABEL('Network Local CA')
RING(KDCRING))
```

f. Give the SKRBKDC permission to read its own key ring by administering a profile in the RDATALIB class.

```
RDEFINE RDATALIB SKRBKDC.KDCRING.LST UACC(NONE)
PERMIT SKRBKDC.KDCRING.LST CLASS(RDATALIB) ID(SKRBKDC) ACCESS(READ)
```

- If the RDATALIB class is not already active, activate and RACLIST it.

```
SETROPTS CLASSACT(RDATALIB) RACLIST(RDATALIB)
```

- If the RDATALIB class is already active and RACLISTed, refresh it.

```
SETROPTS RACLIST(RDATALIB) REFRESH
```

g. Edit /etc/skrb/home/kdc/envar to point to the KDC keyring

```
SKDC_PKINIT_KEYRING=SKRBKDC/KDCRING
```

Note: This set up assumes the client would accept the KDC certificate with the Domain name in the Subject Alternate Name extension, i.e pkinit_kdc_hostname=dcesec4.krb390.ibm.com is specified in the realm section in the krb5.conf file.

KRB2003.IBM.COM configuration

For this configuration, do these steps:

1. The usual Windows® 2003 server installation sets up the active directory and creates SRV records for the **_kerberos** and **_kpasswd** services provided by the Windows 2003 domain controller. However, the server installation does not create a TXT record to map host names in the Windows 2003 domain, so you have to create one yourself.

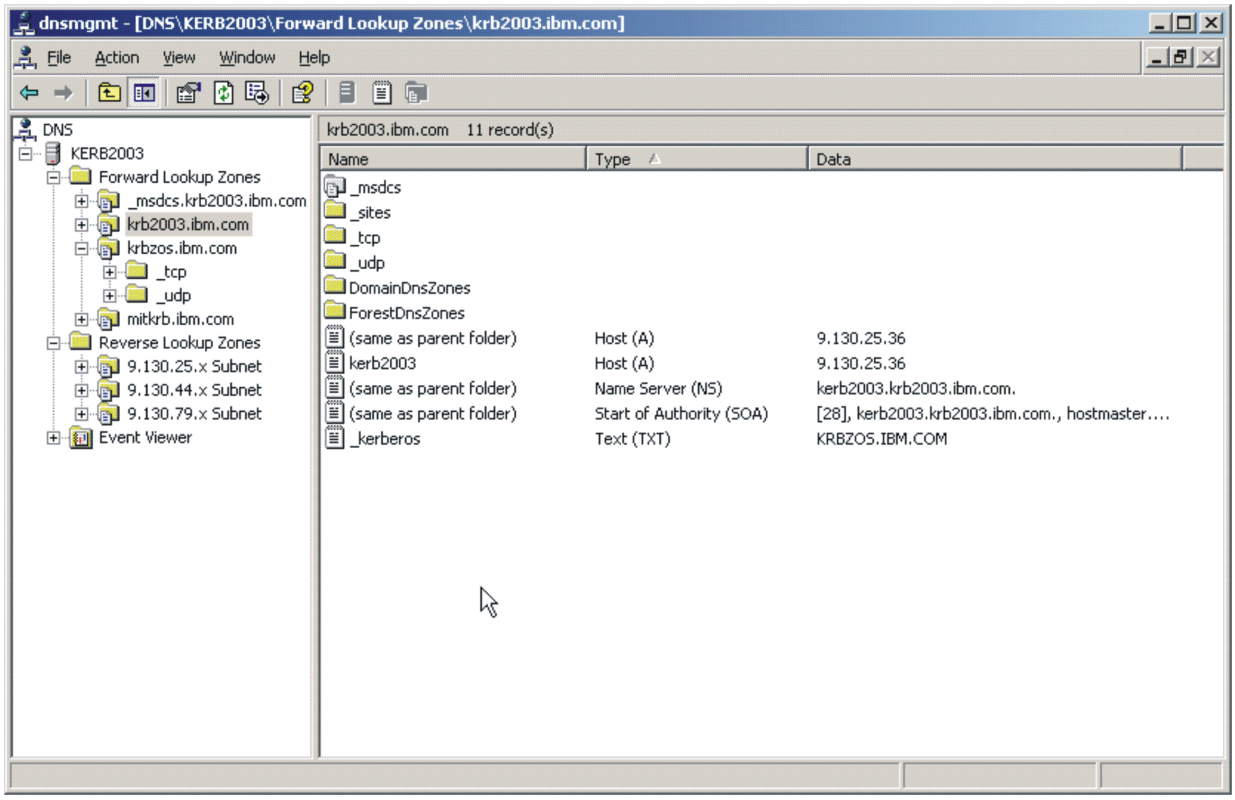

Figure 6. KRB2003.IBM.COM configuration - creating a TXT record to map host names

2. Set up a peer-to-peer trust relationship between the KRB390.IBM.COM realm and the KRB2003.IBM.COM. This allows clients in the KRB2003.IBM.COM realm to access services in the KRB390.IBM.COM realm, and vice versa.
3. The following RACF commands set up the z/OS side of the peer-to-peer trust relationship. Note that the password is case-sensitive on the RDEFINE REALM command but the realm name is converted to uppercase.

```
RDEFINE REALM
    /.../KRB390.IBM.COM/krbtgt/KRB2003.IBM.COM
    KERB(PASSWORD(peerw2kp))
RDEFINE REALM
    /.../KRB2003.IBM.COM/krbtgt/KRB390.IBM.COM
    KERB(PASSWORD(peer390p))
```

4. Windows 2003 does not support the DESD, DES3, AES128, AES256, AES128SHA2, and AES256SHA2 encryption types. If these encryption types have been enabled for the z/OS KDC, they should be disabled for cross-realm ticket-granting tickets issued for the Windows 2003 realm.

```
RALTER REALM
     /.../KRB390.IBM.COM/krbtgt/KRB2003.IBM.COM
     KERB(ENCRYPT(NODESD NODES3 NOAES128 NOAES256 NOAES128SHA2 NOAES256SHA2))
```

5. Use the Active Directory Domains and Trusts management console to set up the Windows 2003 side of the peer-to-peer trust relationships. Open the Properties dialog for the **krb2003.ibm.com** domain. The password specified for the 'Domains trusted by this domain' entry must be the same as the password specified on the /.../KRB390.IBM.COM/krbtgt/KRB2003.IBM.COM RDEFINE command. The password specified for the 'Domains that trust this domain' entry must be the same as the password specified on the /.../KRB2003.IBM.COM/krbtgt/KRB390.IBM.COM RDEFINE command.

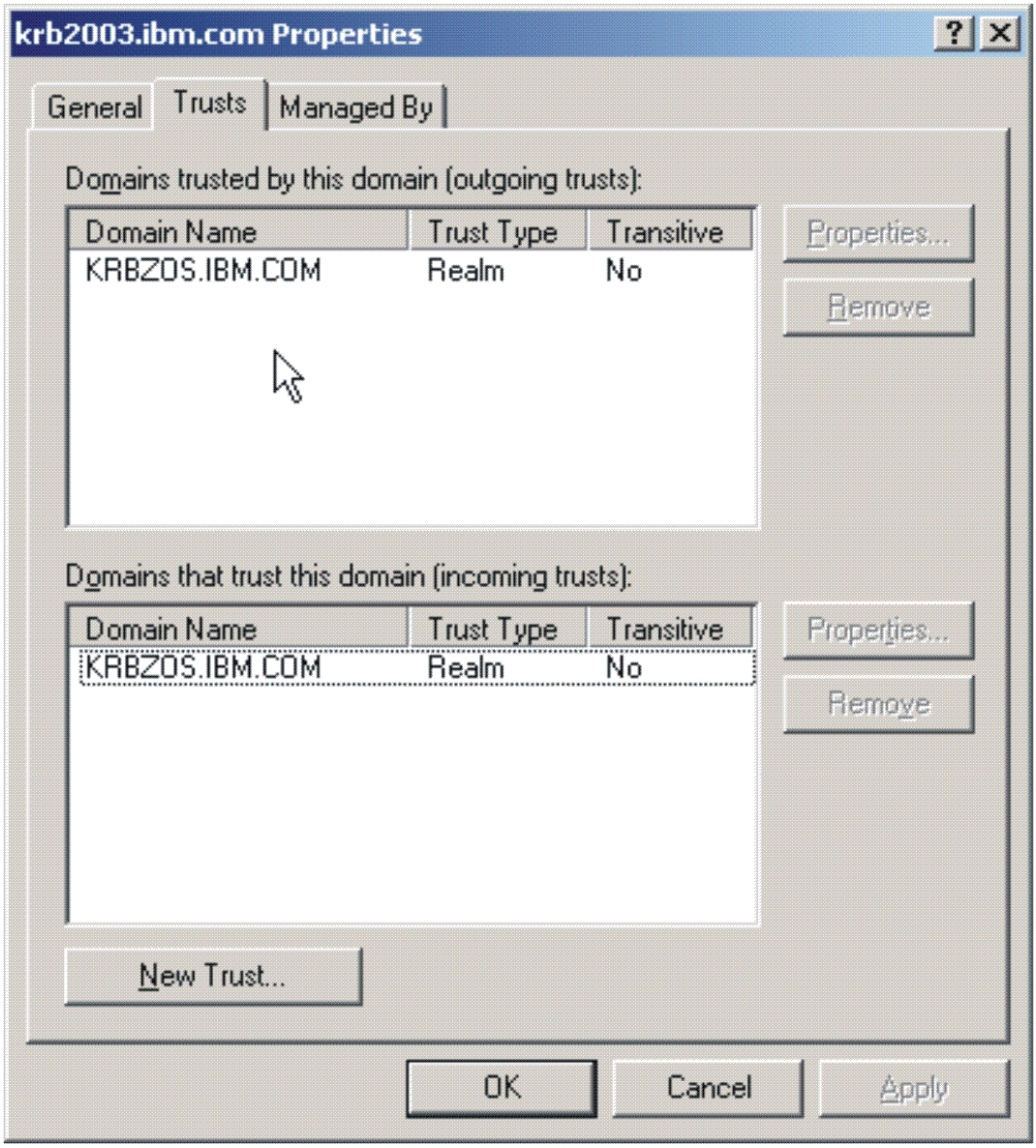

Figure 7. KRB390.IBM.COM configuration - setting up the Windows 2003 side of the peer-to-peer trust relationships

6. Define the location of the KRB390.IBM.COM KDC on each Windows 2003 client using the **ksetup** command. This command is shipped as part of the Windows 2003 Support Tools on the Windows 2003 CD.

```
ksetup /addkdc KRB390.IBM.COM dcesec4.krb390.ibm.com
ksetup /addkdc KRB390.IBM.COM dcesec7.krb390.ibm.com
```

MITKRB.IBM.COM configuration

For this configuration, do these steps:

1. Edit **/etc/krb5.conf** and add definitions for the KRB390.IBM.COM and KRB2003.IBM.COM realms.

```
[libdefaults]
    ticket_lifetime = 600
    default_realm = MITKRB.IBM.COM
    default_tkt_enctypes = des-cbc-crc
    default_tgs_enctypes = des-cbc-crc
    default_keytab_name = /etc/krb5.keytab
    default_tkt_enctypes = aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1-kd
    default_tgs_enctypes = aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1-kd

[realms]
    MITKRB.IBM.COM = {
        kdc = dcecpt.mitkrb.ibm.com:88
        kpasswd_server = dcecpt.mitkrb.ibm.com:464
        admin_server = dcecpt.mitkrb.ibm.com:749
        default_domain = mitkrb.ibm.com
    }

    KRB390.IBM.COM = {
        kdc = dcesec7.krb390.ibm.com:88
        kpasswd_server = dcesec7.krb390.ibm.com:464
        default_domain = krb390.ibm.com
    }

    KRB2003.IBM.COM = {
        kdc = sstone1.krb2003.ibm.com:88
        kpasswd_server = sstone1.krb2003.ibm.com:464
        default_domain = krb2003.ibm.com
    }

[domain_realm]
    .mitkrb.ibm.com = MITKRB.IBM.COM
    .krb390.ibm.com = KRB390.IBM.COM

[capaths]
MITKRB.IBM.COM = {
    KRB390.IBM.COM = .
    KRB2003.IBM.COM = .
}
```

2. Create SRV records for the **_kerberos** and **_kpasswd** services using the UDP protocol (MIT Kerberos does not support the TCP protocol for the **_kerberos** and **_kpasswd** services). Create an SRV record for the **_kerberos-adm** service using the TCP protocol. This example uses the Windows 2003 DNS management console to create the SRV entries for the **mitkrb.ibm.com** domain.

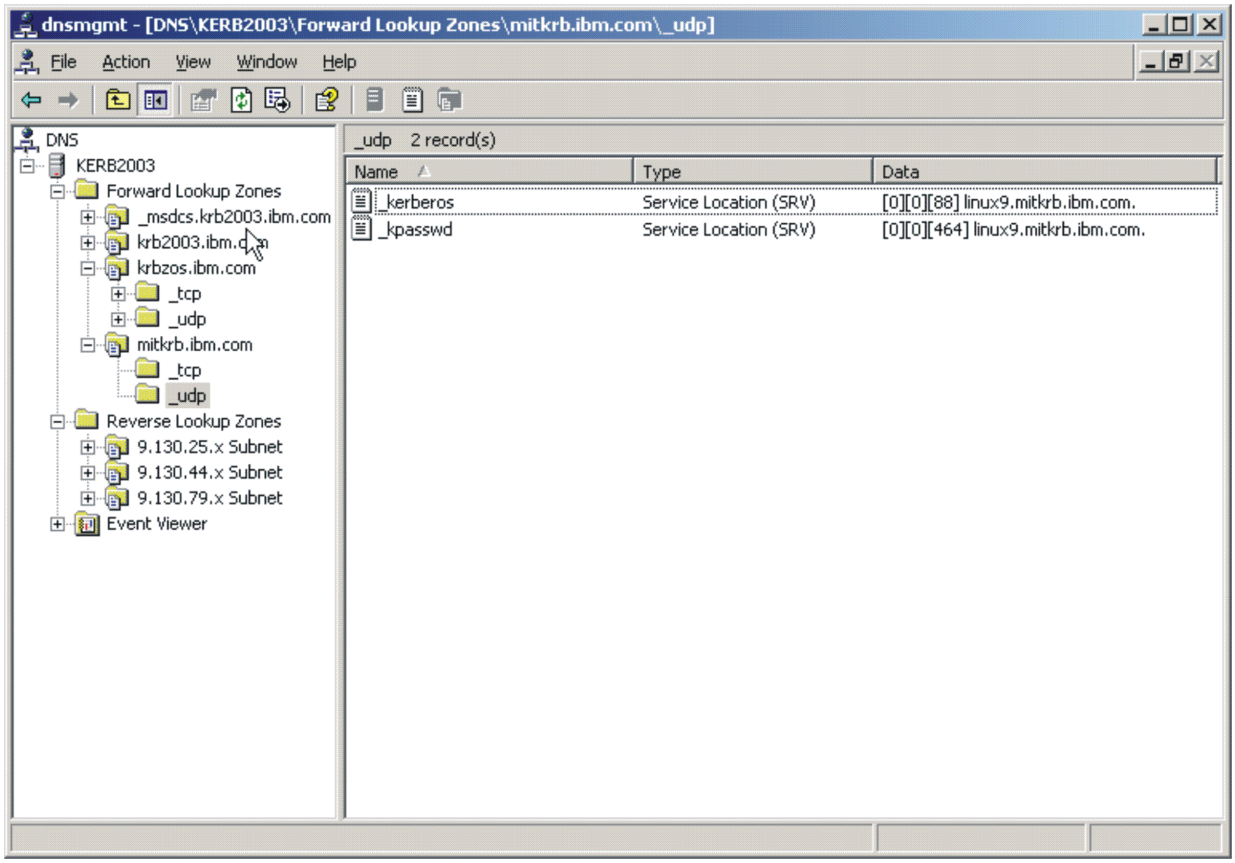

Figure 8. MITKRB.IBM.COM configuration - creating an SRV record using the UDP protocol

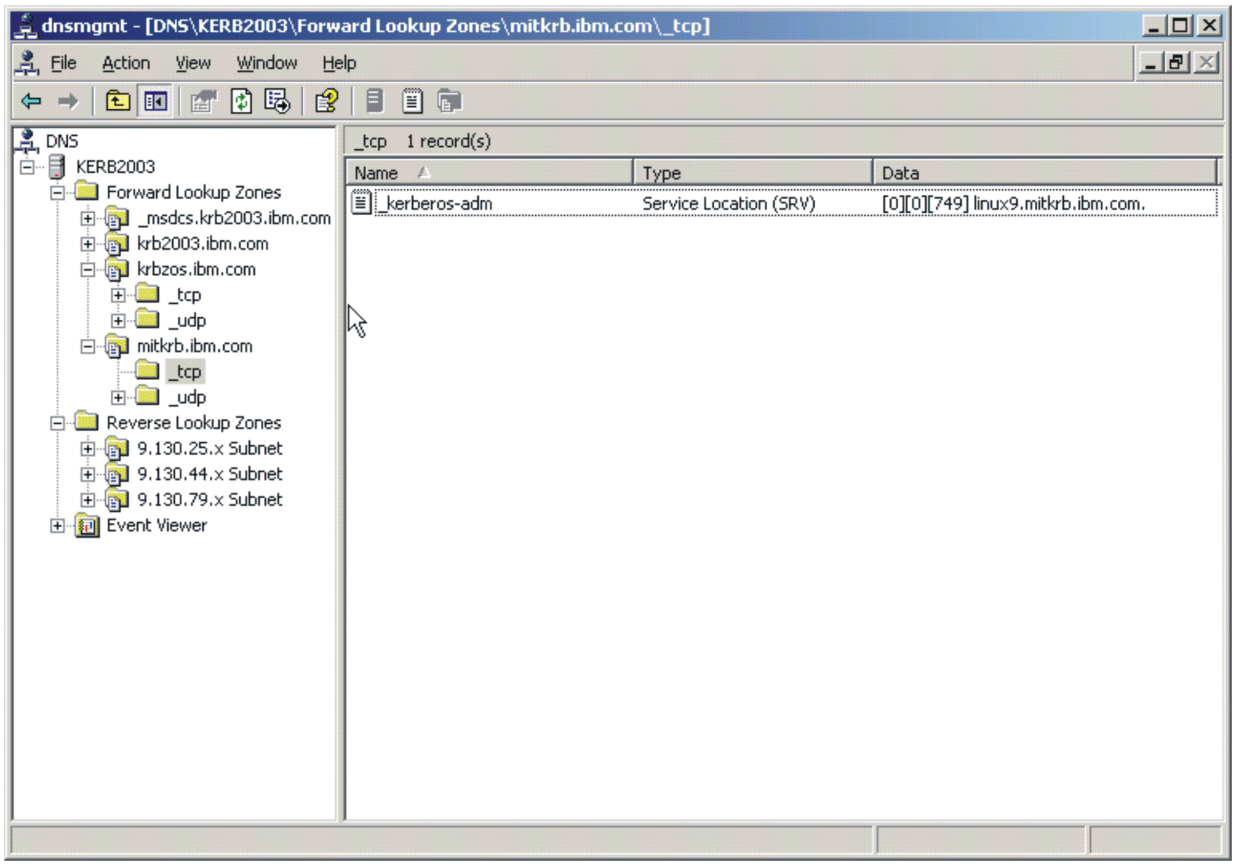

Figure 9. MITKRB.IBM.COM configuration - creating an SRV record using the TCP protocol

3. Create a TXT record to map host names in the **mitkrb.ibm.com** DNS domain to the MITKRB.IBM.COM Kerberos realm. This example uses the Windows 2003 DNS management console to create the TXT record.

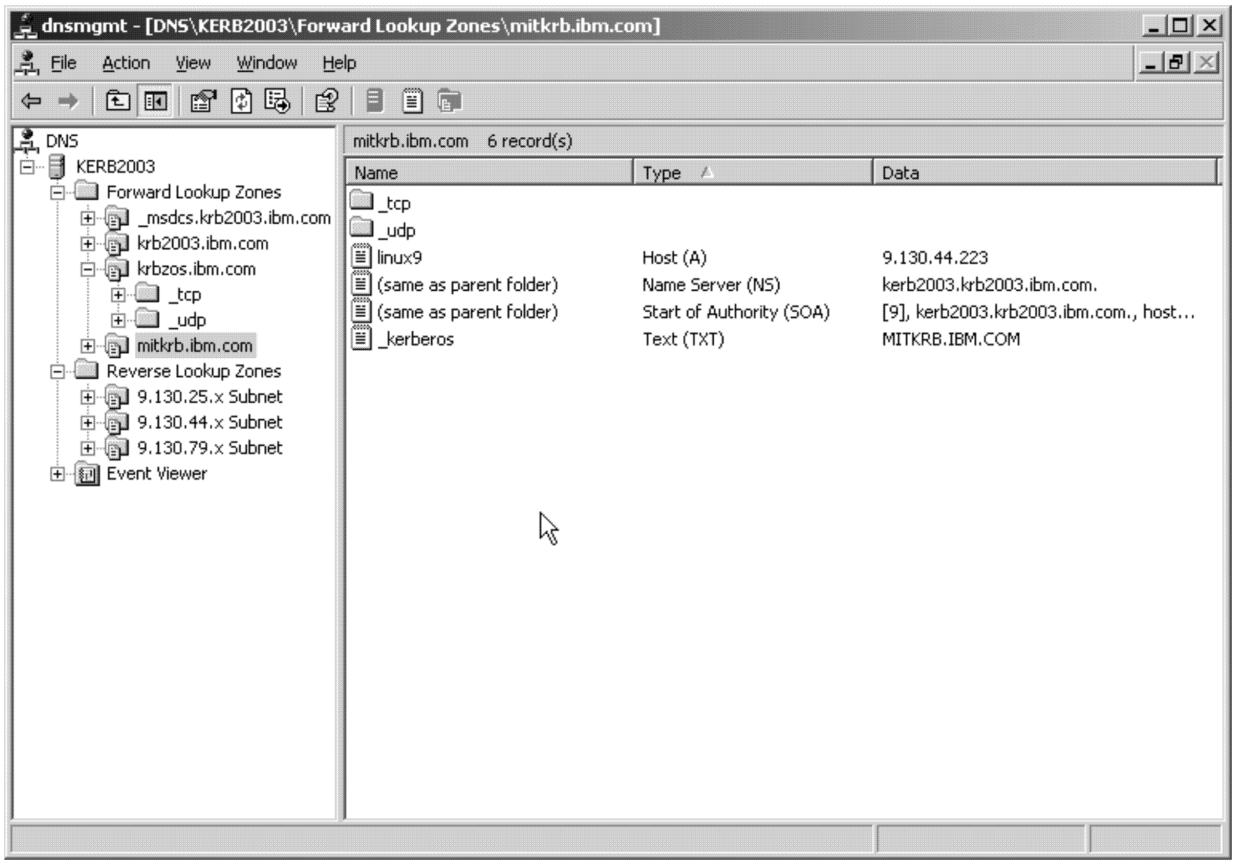

Figure 10. MITKRB.IBM.COM configuration - creating a TXT record to map host names

4. Set up peer-to-peer trust relationships between the KRB390.IBM.COM realm and the MITKRB.IBM.COM and between the KRB2003.IBM.COM realm and the MITKRB.IBM.COM realm. This allows clients in one realm to access services in another realm.
5. The following RACF commands set up the z/OS side of the peer-to-peer trust relationship. Note that the password is case-sensitive on the RDEFINE REALM command but the realm name is converted to uppercase.

```
RDEFINE REALM
    /.../KRB390.IBM.COM/krbtgt/MITKRB.IBM.COM
    KERB(PASSWORD(s3902mit))
RDEFINE REALM
    /.../MITKRB.IBM.COM/krbtgt/KRB390.IBM.COM
    KERB(PASSWORD(mit2s390))
```

6. Use the **kadmin** command to create the peer-to-peer trust relations on the MIT Kerberos side. This example uses the z/OS **kadmin** command to create the **krbtgt** principals in the MITKRB.IBM.COM Kerberos database.

```
DCESEC4:/home/susec4/> kadmin -p rwh/admin@MITKRB.IBM.COM
EUVF06147I Authenticating as rwh/admin@MITKRB.IBM.COM.
EUVF02033R Enter password:

kadmin>
addprinc -pw s3902mit krbtgt/MITKRB.IBM.COM@KRB390.IBM.COM
EUVF06122I Principal krbtgt/MITKRB.IBM.COM@KRB390.IBM.COM added.
kadmin>
addprinc -pw mit2s390 krbtgt/KRB390.IBM.COM@MITKRB.IBM.COM
EUVF06122I Principal krbtgt/KRB390.IBM.COM@MITKRB.IBM.COM added.
kadmin>
addprinc -pw w2k2mit krbtgt/MITKRB.IBM.COM@KRB2003.IBM.COM
EUVF06122I Principal krbtgt/MITKRB.IBM.COM@KRB2003.IBM.COM added.
kadmin>
```

```
addprinc -pw mit2w2k krbtgt/KRB2003.IBM.COM@MITKRB.IBM.COM
EUVF06122I Principal krbtgt/KRB2003.IBM.COM@MITKRB.IBM.COM added.
```

7. Any encryption types that you disable for the MIT KDC must either be disabled for the z/OS KDC or you must disable the cross-realm ticket-granting tickets issued for the MIT Kerberos realm.

```
RALTER REALM
    /.../KRB390.IBM.COM/krbtgt/MITKRB.IBM.COM
    KERB(ENCRYPT(NODESD NODES3))
```

8. Use the Windows 2003 Active Directory Domains and Trusts management console to set up the Windows 2003 side of the peer-to-peer trust relationships. Open the Properties dialog for the **krb2003.ibm.com** domain. The password specified for the 'Domains trusted by this domain' entry must be the same as the password specified for the /.../MITKRB.IBM.COM/krbtgt/KRB2003.IBM.COM principal. The password specified for the 'Domains that trust this domain' entry must be the same as the password specified for the /.../KRB2003.IBM.COM/krbtgt/MITKRB.IBM.COM principal.

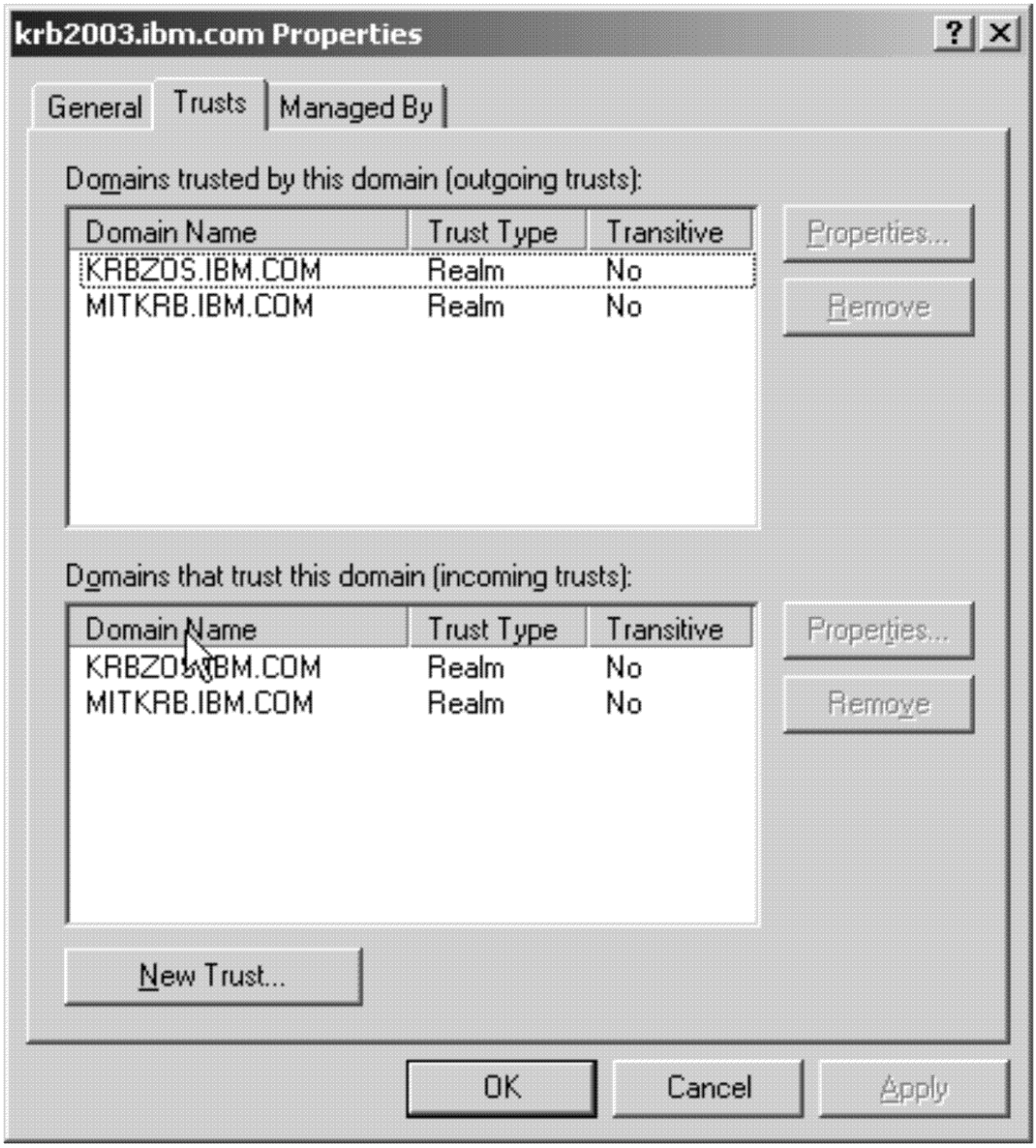

Figure 11. MITKRB.IBM.COM configuration - setting up the Windows 2003 side of the peer-to-peer trust relationships

9. Define the location of the MITKRB.IBM.COM KDC on each Windows 2003 client using the ksetup command. This command is shipped as part of the Windows 2003 Support Tools on the Windows 2003 CD.

```
ksetup /addkdc MITKRB.IBM.COM dcecpt.mitkrb.ibm.com
```

Appendix B. Accessibility

Accessible publications for this product are offered through IBM Documentation (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the Contact the z/OS team web page (www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

This glossary defines technical terms and abbreviations used in the documentation for z/OS Security Server Network Authentication Service.

Authentication
Verifying the claimed identity of a principal.

Authentication header
A record containing a ticket and an authenticator to be presented to a server as part of the authentication process.

Authentication path
A sequence of intermediate realms transited in the authentication process when communicating from one realm to another.

Authenticator
A record containing information that can be shown to have been recently generated using the session key known only by the client and the server.

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.

Ciphertext
The output of an encryption function. Encryption transforms plaintext into ciphertext.

Client
A process that makes use of a network service on behalf of a user. Note that in many cases a server may itself be a client of some other server (for example, a print server may be a client of a file server).

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

KDC
Key Distribution Center, a network service that supplies tickets and temporary session keys; or an instance of that service or the host on which it runs. The KDC processes both initial ticket and ticket-granting ticket requests. The initial ticket portion is sometimes referred to as the Authentication Service (AS) while the ticket-granting portion is sometimes referred to as the Ticket Granting Service (TGS).

Kerberos
This is the name given to the Massachusetts Institute of Technology (MIT) computing system authentication service, the protocol used by that service, and the programs used to implement the authentication service. The name comes from Greek mythology: Kerberos was the 3-headed dog guarding Hades.

Plaintext
The input to an encryption function or the output of a decryption function. Decryption transforms ciphertext into plaintext.

Principal
A uniquely named client or server instance that participates in a network communication.

Principal identifier
The name used to uniquely identify each different principal.

Seal
To encipher a record containing several fields in such a way that the fields cannot be individually replaced without either knowledge of the encryption key or leaving evidence of tampering.

Secret key
An encryption key shared by a principal and the KDC, distributed outside the bounds of the system, with a long lifetime. In the case of a human user's principal, the secret key is derived from a password.

Server
A particular principal that provides a resource to network clients.

Service
A resource provided to network clients, often provided by more than one server.

Session key
A temporary encryption key used between two principals, with a lifetime limited to the duration of a single login session.

Sub-session key
A temporary encryption key used between two principals, selected and exchanged by the principals using the session key, and with a lifetime limited to the duration of a single association.

Ticket
A record that helps a client authenticate itself to a server; it contains the client's identity, a session key, a timestamp, and other information, all sealed using the server's secret key. It serves to authenticate a client only when presented along with a fresh authenticator.

Index

Product Number: 5650-ZOS

SC23-6786-50