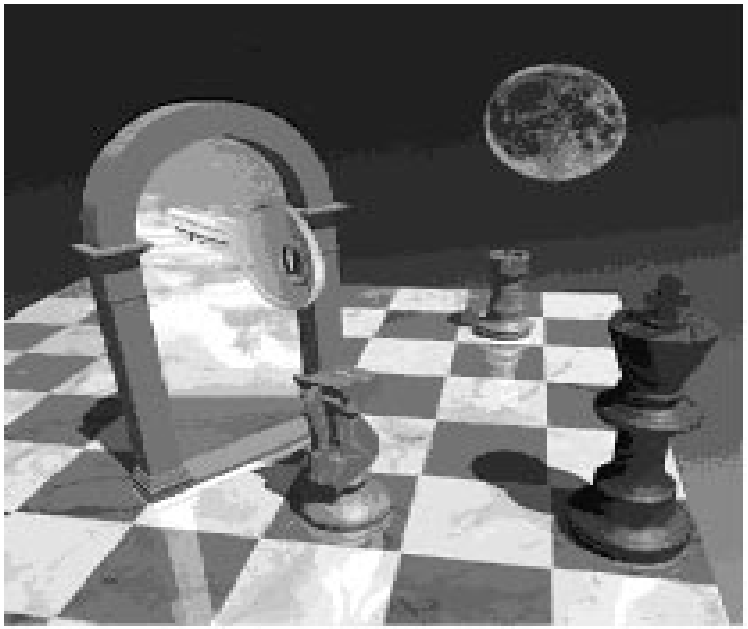
Network Authentication Service Kerberos on z/OS - Part II RACF Implementation

SHARE SESSION 1727
July 24, 2001



Eric Rosenfeld
Security Development
IBM Poughkeepsie
rosenfel@us.ibm.com

Agenda

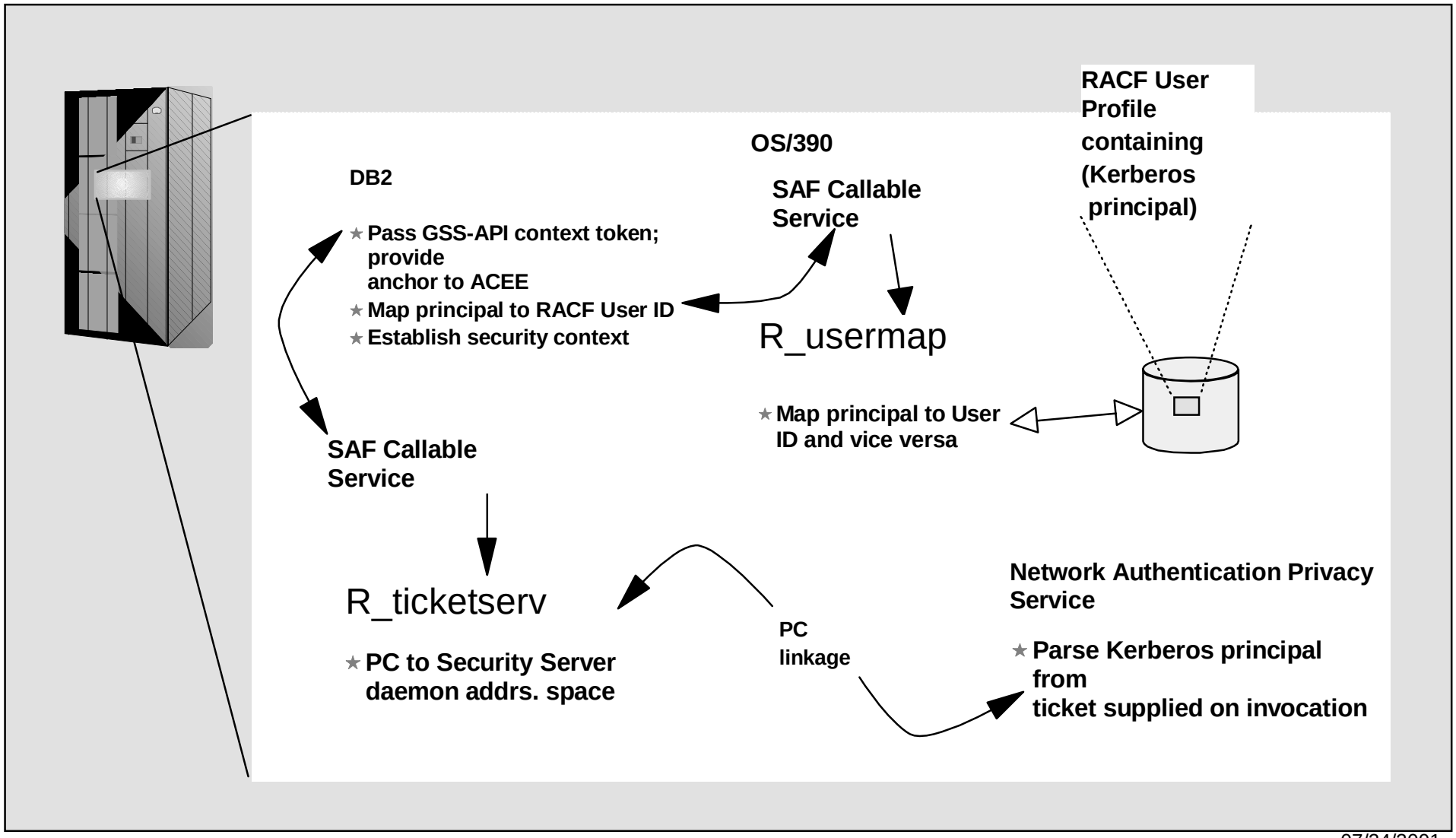
- Kerberos Registry Support Overview
- Getting started
  - Server information
  - Commands
- Callable Services
- Dependencies and Migration Considerations
- Future Support
- Session Summary

Appendices

- Appendix A: Auditing
- Appendix B: Database Unload
- Appendix C: Messages
- Appendix D: Trademarks

OS/390 and NT Kerberos Domains...

This pictorial indicates that OS/390 needs to be a viewed as a Kerberos peer domain. Administratively, a peer trust relationship has been established between the OS/390 Kerberos domain and a Win2K Kerberos domain. Local Kerberos principals must be defined to the OS/390 Security Server and a new user profile segment will hold the Kerberos principal name. Support is also provided to map a Kerberos principal name to a RACF User ID. Note that principal registration must be performed in two places, 1) to the Win2k Kerberos domain, and 2) to the OS/390 Kerberos domain.



Kerberos Server Support

- RACF callable services are enhanced
  - R_usermap
    - Enhanced to support mapping a Kerberos local or foreign principal to a RACF user identity
  - R_admin
    - Enhanced to support the new Kerberos User and General Resource information

Kerberos Server Support...

- New RACF callable Services
  - R_kerbinfo is called by the server to:
    - Retrieve principal information
    - Retrieve realm information
    - Update the count of invalid key attempts
      - similar to an invalid logon attempt
    - Reset the count of invalid key attempts
      - like when you remember your password, on your 2nd or 3rd try
  - R_ticketserv is called by applications to determine the principal name associated with a credential

New Classes

- KERBLINK
  - Maps Kerberos principal to RACF userid
    - ADDUSER/ALTUSER defines local profiles
    - RDEF/RALT used to define foreign profiles
- REALM
  - Defines default information for local realm (KERBDFLT)
  - Defines inter-realm trust

Miscellaneous Support

- The IRRRID00 utility is enhanced to recognize KERBLINK class profiles and flag references to RACF users which do not exist
- The Database unload utility is enhanced to unload new Kerberos user/principal information and new Kerberos realm general resource information
- The SMF unload utility is enhanced to unload SMF records cut by the IBM Kerberos server

Steps for Getting Started

- Install/Customize Network Authentication Server
- Define local realm
- Define inter-realm relationships
- Define local principals
- Define foreign principals

Network Authentication Service - Installation ...

- HFS directories needed
  - /etc/skrb
  - /etc/skrb/home
  - /etc/skrb/home/kdc
  - /var/skrb (needs to be setup !)
- chmod all above to 755
- chmod /var/skrb/creds 777

Network Authentication Service - Installation ...

- Set-up RRSF(RACF Remote Sharing) in local mode
- Create SKRBKDC userid (for the Started Task)
- Activate APPL class if not already active
- Define SKRBKDC application
- Set universal access to READ (if applicable)
- Refresh APPL class

Network Authentication Service - Installation ...

- Define SKRBKDC started task and associate it with SKRBKDC userid
- Refresh STARTED class
- Copy SKRBKDC started task procedure from EUVF.SEUVFSAM to SYS1.PROCLIB
- Copy SKRBKDC environment variables definitions to /etc/skrb/home/kdc/envar
- Set TZ and RESOLVER_CONFIG for your installation

Network Authentication Service - Configuration

- The krb5.conf file - found by env. var. KRB5_CONFIG
  - default is /etc/skrb/krb5.conf
- sample in /usr/lpp/skrb/examples/krb5.conf
  - permissions should be read for everyone, only administrator may modify
  - modified only in code page 1047

Network Authentication Service - Installation

- Kerberos product is installed in HFS
  - /usr/lpp/skrb
- System dataset changes
  - Add EUVF.SEUVFLPA to LPALST
  - Add EUVF.SEUVFLNK to LNKLST
  - Add EUVF.SEUVFEXC to SYSEXEC DD concatenation for TSO

Realm Commands

- Realm definition with RDEFINE/RALTER
  - Realm class profile
  - Ticket life values
    - DEFTKTLFE - default ticket life
    - MAXTKTLFE - maximum ticket life
    - MINTKTLFE - minimum ticket life
    - Only valid for local realm
    - If one is specified all three values must be for RDEFINE
    - All three values must be on command or in DB for RALTER
    - Range from 1 to 2,147,483,647 seconds

Realm Commands *(continued)*

- **KERBNAME** - unqualified name of the local Kerberos realm
  - Max length of 117 characters
  - Can not contain '/'
  - EBCDIC variant characters should not be used
- **PASSWORD** - realm password
  - Max length of 8 characters
  - EBCDIC variant characters should not be used
- **NODEFTKTLFE, NOMAXTKTLFE, NOKERBNAME, NOMINTKTLFE, NOPASSWORD**, and **NOKERB** only for RALTER

Realm Commands *(continued)*

- Profile naming
  - Defining a local realm
    - Profile name must be KERBDFLT
    - KERBNAME field has unqualified local realm name
    - Realm name is rolled to upper case
  - Defining an inter-realm trust relationship
    - Can consist of two REALM class profiles
      - Profile name: /.../LOCAL_REALM/krbtgt/REALM_2
        - krbtgt/REALM_2@LOCAL_REALM
      - Profile name: /.../REALM_2/krbtgt/LOCAL_REALM
        - krbtgt/LOCAL_REALM@REALM2

Realm Command *Examples*

- Local Realm example:
  - RDEFINE REALM KERBDFLT KERB(KERBNAME(KRB390.IBM.COM) PASSWORD(xxxx) MINTKTLFE(15) DEFTKTLFE(36000) MAXTKTLFE(86400))
- Inter-realm trust example:
  - RDEFINE REALM /.../KRB390.IBM.COM/krbtgt/KRB2000.IBM.COM KERB(PASSWORD(password))
  - RDEFINE REALM /.../KRB2000.IBM.COM/krbtgt/KRB390.IBM.COM KERB(PASSWORD(password))

User Commands

- Local principal definition with ADDUSER/ALTUSER
  - Local realm must exist before issuing command
  - **MAXTKTLFE** specifies the local principal maximum ticket life
  - **KERBNAME** is the unique name of a local principal.
    - Can not contain '@'
    - Variant characters should not be used
    - Can not exceed 240 characters when fully qualified with the local realm name
      - /.../local_realm/kerbname_1
    - Must be entered unqualified
  - **NOMAXTKTLFE, NOKERBNAME, NOKERB** only valid on ALTUSER
  - Kerberos keys generated at first non-expired password setting
  - KERBLINK mapping profile created/updated

LISTUSER - Key information

When the initial KERB segment is added via

ADDUSER USER1 KERB(KERBNAME(User1))

the password is not yet synchronized with the Kerberos local principal's password:

LISTUSER USER1 KERB NORACF

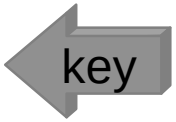
USER=USER1
KERB INFORMATION

KERBNAME= User1

After a password change, the key is generated !

USER=USER1
KERB INFORMATION

KERBNAME= User1
KEY VERSION= 001



Mapping Foreign Users

- Foreign Kerberos principals are mapped to a RACF identity using KERBLINK class profiles
- RDEFINE KERBLINK /.../foreign_realm/foreign_principal APPLDATA('racf_user')
  - Maps single foreign principal to a RACF userid
- RDEFINE KERBLINK /.../foreign_realm/ APPLDATA('racf_user')
  - Maps all principals for a single realm to a RACF userid
- Realm names are rolled to upper case

SETROPTS Command

- Special case logic added to prevent the explicit or implicit activation of generic profile checking and generic command processing for the KERBLINK and REALM classes
- SETR GENERIC(KERBLINK REALM) GENCMD(KERBLINK REALM) will result in a new message
- SETR GENERIC(*) GENCMD(*) will **ignore** the KERBLINK and REALM classes

Steps for Getting Started

- Install/Customize Server
- Define local realm
  - RDEFINE REALM KERBDFLT KERB(KERBNAME(realm) PASSWORD(realmpass))
- Define inter-realm relationship
  - RDEFINE REALM /.../realm1/krbtgt/realm2 KERB(PASSWORD(TrustP1))
  - RDEFINE REALM/.../realm2/krbtgt/realm1 KERB(PASSWORD(TrustP2))
- Define local principals
  - ALTUSER user1 KERB(KERBNAME(KerbUSER1)) PASSWORD(usrp) NOEXPIRED
- Define foreign principals
  - RDEFINE KERBLINK /.../foreign_realm/foreign_principal APPLDATA('racf_user')
  - RDEFINE KERBLINK /.../foreign_realm/ APPLDATA('racf_user')

Callable Service: R_usermap (IRRSIM00)

- Map application user
  - The following function codes were added:
    - UMAP_R_TO_K (5) -- return the Kerberos application user identity for the supplied RACF user ID
    - UMAP_K_TO_R (6) -- return the RACF user ID associated with the supplied Kerberos application user identity

Callable Service: R_ticketserv (IRRSPK00)

- Parse or extract Kerberos principal
  - Function code
    - TKTS_RETURN_NAME (1) - Parse specified ticket and return Kerberos principal name
      - GSS-API context token is input
      - Principal name is output

Callable Service: R_admin (IRRSEQ00)

- Support added for
  - ADMN_ADD_USER, ADMN_ALT_USER, ADMN_LST_USER ADMN_ADD_GENRES, ADMN_ALT_GENRES, ADMN_LST_GENRES to support KERB segment fields
- New fields
  - KERBNAME - realm or principal name
  - MAXTKTLF - realm or principal maximum ticket life
  - MINTKTLF - realm wide minimum ticket life
  - DEFTKTLF - realm wide default ticket life
  - PASSWORD - realm password

Dependencies and Migration Considerations

- OS/390 SecureWay Network Authentication Service server will use the profile definitions in the KERB segments of users and realm definitions via R_kerbinfo
- Any application can use R_ticketserv and R_usermap to map Kerberos information to RACF
- Migration and Coexistence
  - RRSF local node must be defined to allow for keys to be generated for user password application updates
  - Only password changes from Kerberos aware systems will cause the generation of keys

How do I get this support?

- SecureWay Network Authentication Service server (HSWK2A0)
- OS/390 and RACF R10 (HBB7703, HRF7703)

  or

  PTFs on OS/390 and RACF
  - UW72456 SAF R8 (HBB6608)
  - UW72457 SAF R9 (JBB6609)
  - UW72458 RACF R8 (HRF2608)

What's Coming?

RACF Kerberos Extensions

- Allow more encryption types for keys
  - DES
  - DES3
  - DES with Derivation
  - Allow/disallow each type on a per profile basis
  - Enabled via AU/ALU RDEF/RALT
- New support activated by SETROPTS command KERBLVL setting

Command Keyword Updates

- ENCRYPT(DES|NODES DES3|NODES3 DESD|NODESD)
  - Allowed on RDEFINE/RALTER and ADDUSER/ALTUSER
- KERBLVL(0|1)
  - Added to SETROPTS command
    - 0 - Process at original level of support
    - 1 - Incorporate multiple key functions

Other Updates

- R_kerbinfo
  - Returns the bits associated with allowable encryption types for this profile
  - Key values return in new three key format
- Database Templates
  - New ENCRYPT field defined on user and general resource profiles
- Dynamic Parse
  - ENCRYPT keyword added
    - Valid settings: DES|NODES DES3|NODES3 DESD|NODESD

Migration Considerations

- The proper level of Network Authentication Service server must be installed prior to defining any keys

- SETROPTS KERBLVL setting
  - 0 (Default R10/PTF support level)
  - 1 (Multiple key support active)
  - Do not upgrade to level 1 until all systems sharing the DB have multiple key code level
  - Can set ENCRYPT values at either level, but has no effect until KERBLVL set to 1

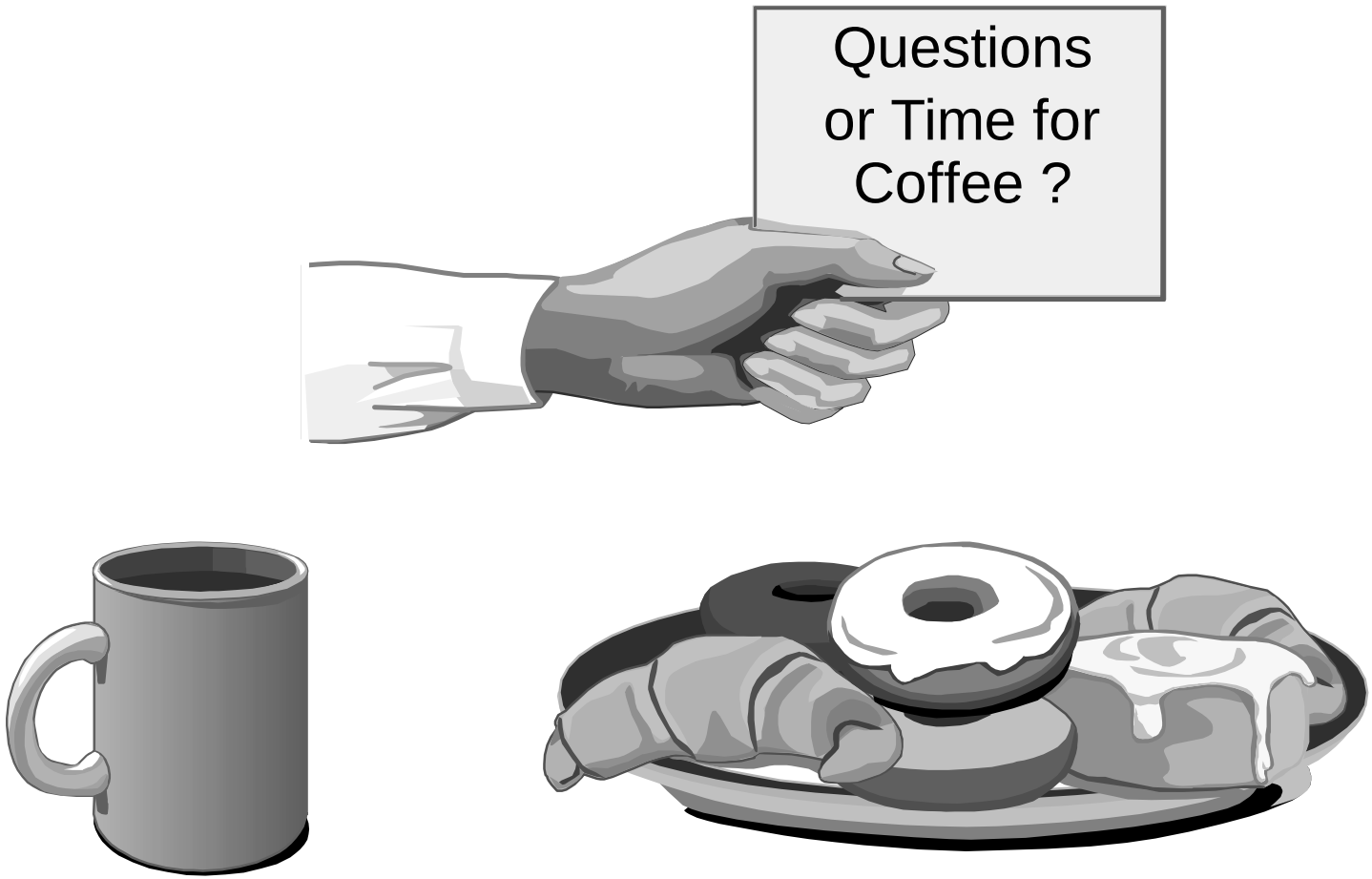
Session Summary

- What we have covered:
  - How RACF interacts with the SecureWay Network Authentication Service on OS/390
  - How an application would interact with RACF to map Kerberos constructs to RACF constructs
  - Migration requirements for the installation of RACF support for Kerberos
  - A glimpse of future support

Publications

- **GC28-1921-07 OS/390 SecureWay Security Server (RACF) Callable Services**
- **SC28-1919-07 OS/390 SecureWay Security Server (RACF) Command Language Reference**
- **SY27-2640-07 OS/390 SecureWay Security Server (RACF) Data Areas**
- **SY27-2639-07 OS/390 SecureWay Security Server (RACF) Macros and Interfaces**
- **SC28-1918-07 OS/390 SecureWay Security Server (RACF) Messages and Codes**
- **GC28-1920-07 OS/390 SecureWay Security Server (RACF) Migration**
- **SC28-1915-07 OS/390 SecureWay Security Server (RACF) Security Administrator's Guide**

Questions ???



Appendix A: Auditing

- Changes to SMF records
  - The SecureWay Kerberos Security Server will cut SMF Type 80 records
    - Event code 68 (X'44') -- grant of initial Kerberos ticket -- reserved for SecureWay Kerberos
      - Event code qualifier 0 -- success
      - Event code qualifier 1 -- failure
    - Extended length relocate section data types -- reserved for SecureWay Kerberos
      - 333 -- Kerberos principal name
      - 334 -- Kerberos login request source
      - 335 -- Kerberos KDC status code
  - SMF Data Unload (IRRADU00) will unload the SecureWay Kerberos Security Server records

Auditing

- Changes to SMF Data Unload

New auditable event introduced for Kerberos

Event Code Name	Column ID	Event Code Number	Description
KTICKET	KTKT	68	Grant of initial Kerberos ticket

Format of Kerberos ticket record extension

Field Name	Type	Length	Start	End	Comments
KTKT_PRINCIPAL	Char	240	282	521	Principal name
KTKT_LOGIN_SOURCE	Char	22	523	544	Login request source
KTKT_KDC_STAT_CODE	Char	10	546	555	KDC status code

Auditing (New level)

- Changes to SMF Data Unload

New field in Type 81 record for system options

Field Name	Type	Length	Start	End	Comments
RINI_KERBLVL	CHAR	4	709	712	Kerberos Processing Level

Appendix B: Database Unload (IRRDBU00)

New User record

Record Name	Record Type	Record Prefix
User KERB Data	02D0	USKERB

User KERB Data Record

Field Name	Type	Start	End	Comments
USKERB_RECORD_TYPE	Int	1	4	Record type (02D0)
USKERB_NAME	Char	6	13	RACF user name
USKERB_KERBNAME	Char	15	254	Kerberos principal name
USKERB_MAX_LIFE	Int	256	265	Maximum ticket life
USKERB_KEY_VERS	Int	267	269	Current key version

Database Unload (IRRDBU00) New level

User record

Record Name	Record Type	Record Prefix
User KERB Data	02D0	USKERB

New Fields: User KERB Data Record

Field Name	Type	Start	End	Comments
USKERB_ENCRPT_DES	YES/NO	271	274	DES enabled
USKERB_ENCRPT_DES3	YES/NO	276	279	DES3 enabled
USKERB_ENCRPT_DESD	YES/NO	281	284	DESD enabled

Database Unload (IRRDBU00)

New General Resource record

Record Name	Record Type	Record Prefix
General Resource KERB Data	0580	GRKERB

General Resource KERB Data Record

Field Name	Type	Start	End	Comments
GRKERB_RECORD_TYPE	Int	1	4	Record type (0580)
GRKERB_NAME	Char	6	251	General resource name
GRKERB_CLASS_NAME	Char	253	260	Name of the class
GRKERB_KERBNAME	Char	262	501	Kerberos realm name
GRKERB_MIN_LIFE	Int	503	512	Minimum ticket life
GRKERB_MAX_LIFE	Int	514	523	Maximum ticket life
GRKERB_DEF_LIFE	Int	525	534	Default ticket life
GRKERB_KEY_VERS	Int	536	538	Current key version

Database Unload (IRRDBU00) New Level

General Resource record

Record Name	Record Type	Record Prefix
General Resource KERB Data	0580	GRKERB

New Fields: General Resource KERB Data Record

Field Name	Type	Start	End	Comments
GRKERB_ENCRPT_DES	YES/NO	540	543	DES enabled
GRKERB_ENCRPT_DES3	YES/NO	545	548	DES3 enabled
GRKERB_ENCRPT_DESD	YES/NO	550	553	DESD enabled

Appendix C: New Messages

ID	TEXT	WHERE ISSUED
ICH08016I	**ERROR SETTING KERBEROS KEY INFORMATION**	**ICHCPA00**
ICH14075I	**SETROPTS** *keyword* **had no effect on class** *classname*.	**ICHCOP06**
IRRC038I	**A request to process Kerberos key information for user** *user* **failed. Processing continues.**	**IRRPWS00**
IRR52162I	**Unable to determine the name of the local Kerberos realm. Command processing ends.**	**ICHCDX29**
IRR52163I	**The "***char***" character is not allowed in KERBNAME. Command processing ends.**	**ICHCDX29**
IRR52164I	**KERBNAME may not be prefixed by "/.../". Command processing ends.**	**ICHCDX29**
IRR52165I	**The value for** *segment_name* **segment** *operand_name* **operand must be unique. Command processing ends.**	**ICHCDX23, ICHCDX24, ICHCDX29**
IRR52166I	**The fully qualified form of the local Kerberos principal must not exceed 240 characters. Command processing ends.**	**ICHCDX29**
IRR52167I	**Unable to validate MINTKTLFE, MAXTKTLFE, or DEFTKTLFE. Ticket lifetime values are ignored.**	**ICHCDX28**
IRR52168I	**Values specified for MINTKTLFE, MAXTKTLFE, or DEFTKTLFE are not valid. Ticket lifetime values are ignored.**	**ICHCDX28**
IRR52169I	**A request to process Kerberos key information for** *profile-name* **failed. Command processing continues.**	**IRRPRE00**

Changed Messages

ID	TEXT	WHERE ISSUED
IRR52151I	Unexpected RACROUTE REQUEST=EXTRACT error while retrieving profile *profile*. SAF RC = X'*safrc*', RACF RC = X'*racfrc*', RACF RSN= X'*rsncode*'.	IRRPRE00
IRR52153I	Unexpected return code *return-code* and reason code *reason-code* encountered while attempting an ICHEINTY operation.	IRRPRE02
IRR52154I	Information in the *class1* mapping profile *profile1* does not match the *profile2* profile in the *class2* class.	IRRPRE00, IRRPRE02

Descriptive text changed to add Kerberos scenarios

Appendix D: Trademarks

- **The following are Trademarks of IBM Corporation**
  - **CICS, DB2, IBM, OS/390, RACF**
- **Windows is a Trademark of Microsoft Corporation**
- **Kerberos is a Trademark of MIT**
- **Other company product or service names may be Trademarks or Servicemarks of others.**