Kerberos on z/OS

SecureWorld Session R05

Network Authentication Service

and

Resource Access Control Facility

Walt Farrell
z/OS Security Development
wfarrell@us.ibm.com

August 2002

Agenda

- General Kerberos Overview
- Kerberos Registry Support Overview
- Getting Started
  - ►Server Information
  - ►Registry set-up
- SAF Callable Services
- Dependencies and Migration Considerations
- z/OS V1R2 extensions
- z/OS V1R4 extensions
- Session Summary

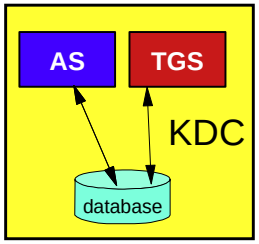
Trademarks

What is Kerberos?

- A distributed authentication service developed by MIT
- Allows user authentication over a physically untrusted network
- Tickets are issued by a Kerberos authentication server: both users and servers are required to have keys registered with the authentication server
- Flows to and from the authentication server establish a session key, used in a direct exchange between a user and a service
- Implemented in Win2K, Solaris 8, OS/390 & z/OS, AIX, OS/400, and more

Key Distribution Center (KDC)

- Trusted "third party"
- Responsible for issuing user credentials and tickets
- Consists of
  - an authentication server (KAS)
  - a ticket granting server (TGS)
  - a Kerberos Data Base (KDB)
    - Contains keys for each user and server



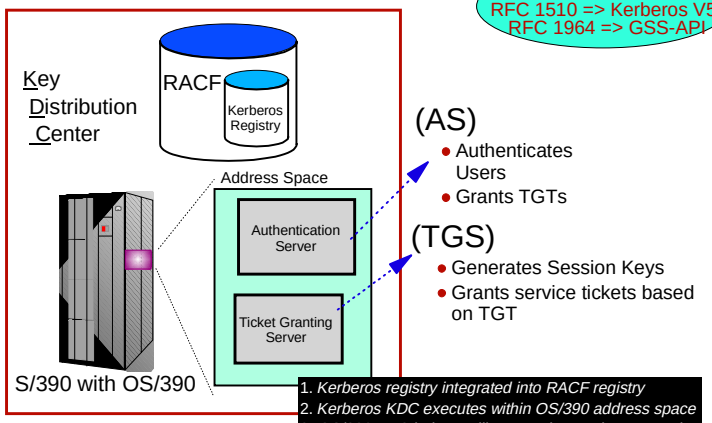
Additional Terms

- Ticket
  - An encrypted electronic authentication token including:
    - client's identity
    - a dynamically created session key
    - a time stamp
    - lifetime for the ticket
    - a service name
- Realm
  - The Kerberos domain: the set of entities which authenticate using the domain of authority served by one KDC.
- Principal
  - Anything that is defined to a realm
  - *name@realm*
    - Can be a user, service or relationship

Kerberos on z/OS and OS/390

(Its own component, intergrated with RACF via SAF)

Standards
RFC 1510 => Kerberos V5
RFC 1964 => GSS-API

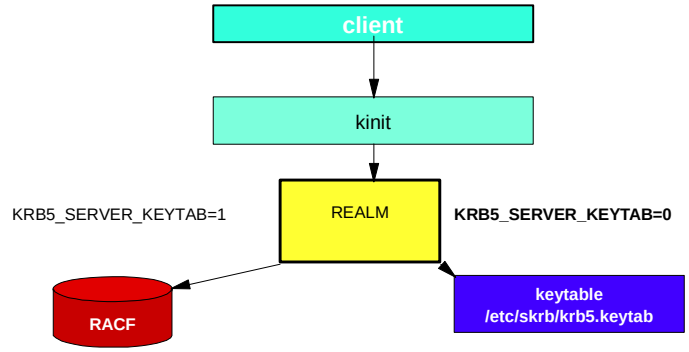


(AS)
- Authenticates Users
- Grants TGTs

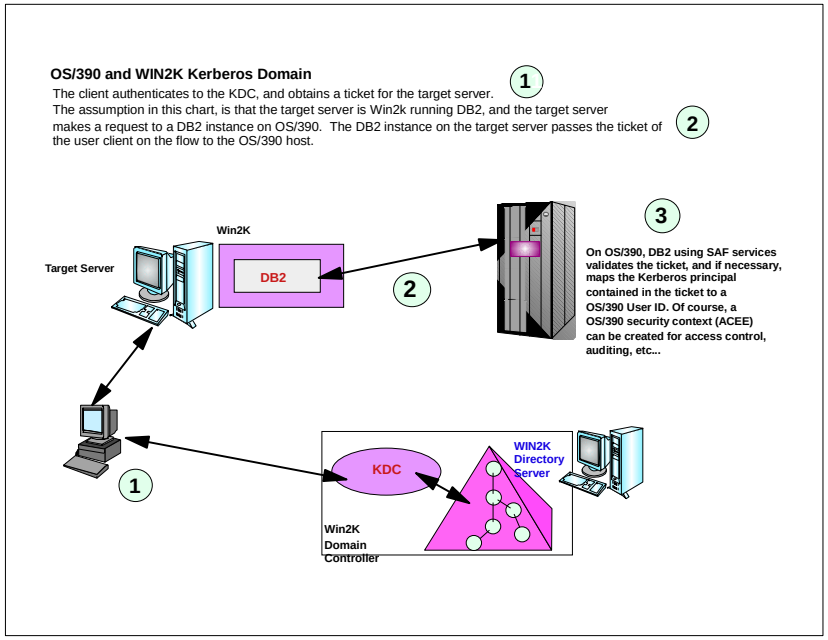
(TGS)
- Generates Session Keys
- Grants service tickets based on TGT

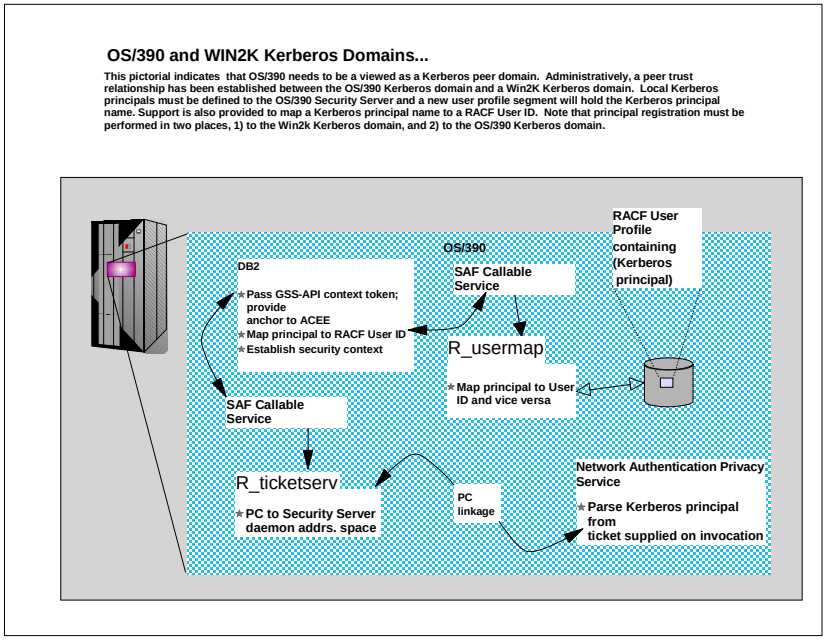
1. *Kerberos registry integrated into RACF registry*
2. *Kerberos KDC executes within OS/390 address space*
3. *OS/390 KDC behaves like any other Kerberos "Realm"*
4. *Kerberos Realm to Realm function supported*

Network Authentication Service - keytab or RACF



User/application needs read access to IRR.RUSERMAP





Network Authentication Service - Commands

- Network Authentication Service provides the standard Kerberos commands provided by Version 5:
  - kinit - obtains or renews the Kerberos ticket-granting ticket.
  - klist - displays the contents of a Kerberos credentials cache or key table.
  - keytab - manages a key table (OS/390 likely will use RACF).
  - kdestroy - destroys a Kerberos credentials cache.
  - ksetup - manages Kerberos service entries in the LDAP directory for a Kerberos realm.

RACF is the Kerberos Registry

- The OS/390 SecureWay Network Authentication Server requires a registry of principal information, global information, etc.
- This security information is stored in RACF User and General Resource profiles
- Kerberos administration is done via RACF commands/panels
- The SecureWay Network Authentication Server obtains it's registry information via SAF callable service
- Kerberos application servers can use SAF callable services to parse Kerberos tickets to obtain principal names, and to map from principal to RACF user and vice versa

RACF as the Kerberos Registry

- Fosters direct interoperation between OS/390 and Kerberos servers and clients
- Places all registry information in the RACF database with its inherent security and integrity
- Allows applications to leverage RACF access control and auditing with distributed user identities
- User password rules are in force for user principal's key definition
- Extends existing administration interfaces and limits new interfaces
- Minimal learning curve for administration changes

Kerberos Registry

- RACF commands/panels are used for administration
  - Local Kerberos principals are defined as RACF users with a KERB segment
  - REALM class profiles are used to define information about the local Kerberos realm and foreign realms
    - Local realm information includes name, key, and ticket lifetime (MIN, MAX, and DEFAULT in seconds)
    - Foreign realm trust relationships are defined in pairs (A to B and B to A) which also include a key
  - Foreign Kerberos principals are mapped to a RACF identity using KERBLINK class profiles

Kerberos Registry

- The RACF user password and the Kerberos local principal's password are integrated
  - Kerberos key will be generated when the user's password changes and is **not** expired
    - TSO/application logon
    - ALU NOEXPIRED
    - PASSWORD command
  - The Kerberos password is subject to RACF SETROPTS rules and installation defined rules via password exit

Kerberos Registry

- 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 Registry

- 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

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

Steps for Getting Started

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

Network Authentication Service - Installation

- Installs into
  - HFS
    - executables in directory /usr/lpp/skrb
    - /etc/skrb files need access 755
    - /var/skrb/creds needs access 777
  - System datasets
    - Add EUVF.SEUVFLPA to LPALST
    - Add EUVF.SEUVFLNK to LNKLST
    - Add EUVF.SEUVFEXC to SYSEXEC DD concatenation for TSO

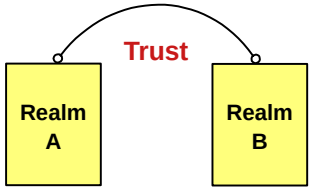
Network Authentication Service - Installation

- Configuration in krb5.conf file
  - KRB5_CONFIG environment variable
  - 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 ...

- Set-up RRSF(RACF Remote Sharing) in local mode
- Define SKRBKDC application and USERID as started task
- Copy SKRBKDC environment variables definitions to /etc/skrb/home/kdc/envar
- Set TZ and RESOLVER_CONFIG for your installation

Registry Definitions



Commands must be entered to define:

A local realm
Inter-realm trust relationships (between KDCs)
Local and foreign principals

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 ...

- **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 ...

- 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 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')
    - maps single principal to a RACF user
  - RDEFINE KERBLINK /.../foreign_realm/ APPLDATA('racf_user')
    - Maps all principals for a single realm to a RACF userid

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

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

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

- Network Authentication Service implements V5 standard
- The IBM Kerberos server requires R_kerbinfo SAF support
- 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)

Kerberos Server Extensions

- Strong Crypto support
- New commands
  - kpasswd - change principal password
  - kvno - Query key version number
  - kadmin - administer KDC via sub-commands
    - help, list_principals, get_principal, add_principal, delete_principal, modify_principal, change_password, rename_principal,etc.
- New Kerberos and GSSAPI APIs
- New console DISPLAY commands
  - XCF, CRYPTO, LEVEL

RACF Kerberos Extensions

- Allow more encryption types for keys
  - DES
  - Triple DES
  - 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

Migration Considerations

- The V1R2 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

How do I get this support?

- SecureWay Network Authentication Service server (HSWK320)
- z/OS and RACF V1R2 (HBB7705, HRF7705)

Kerberos Server Extensions

- Support for IPV6 Network Addressing
- NDBM Support
  - Supports KDC database in UNIX file system instead of RACF database
  - Better interoperability with other platforms
    - remote administration via kadmin
    - database propagation
  - However:
    - May not scale as well as RACF database configuration for large numbers of users
    - SYSPLEX support not as robust
    - May require administering users in both RACF and NDBM

Network Authentication Service - Exploitation

Who uses the Network Authentication Service?

Customers with network-based applications that use Kerberos authentication

IBM products such as:

DB2 V7 / DB2 Connect V7.1 FP2
WebSphere V4 (OS/390 or z/OS)
z/OS V1R2 FTP Client/Server
z/OS V1R2 LDAP Client/Server
z/OS V1R2 Telnet Server
z/OS V1R2 RSH Server

Session Summary

- What we have covered:
  - How RACF interacts with the Network Authentication Service
  - How an application would interact with SAF to map Kerberos constructs to RACF constructs
  - Migration requirements for the installation of Kerberos support
  - An overview of follow-on support

Publications

IBM Books

- **GC28-1921 OS/390 SecureWay Security Server (RACF) Callable Services**
- **SC28-1919 OS/390 SecureWay Security Server (RACF) Command Language Reference**
- **SY27-2640 OS/390 SecureWay Security Server (RACF) Data Areas**
- **SY27-2639 OS/390 SecureWay Security Server (RACF) Macros and Interfaces**
- **SC28-1918 OS/390 SecureWay Security Server (RACF) Messages and Codes**
- **GC28-1920 OS/390 SecureWay Security Server (RACF) Migration**
- **SC28-1915 OS/390 SecureWay Security Server (RACF) Security Administrator's Guide**
- **SC24-5896 OS/390 SecureWay Security Server Network Authentication and Privacy Service Administration**
- **SC24-5897 OS/390 SecureWay Security Server Network Authentication and Privacy Service Programming**

RFCs

- **RFC 1510 - The Kerberos Network Authentication Service (V5)**
- **RFC 1964 - The Kerberos Version 5 GSS-API Mechanism**
- **RFC 2078 - Generic Security Service Application Program Interface (V2)**
- **RFC 2744 - Generic Security Service Application Program Interface (V2): C Bindings**

QUESTIONS ?