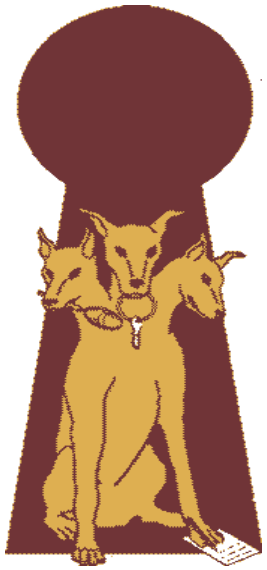
19th Annual Vanguard
Enterprise Security Expo™
& RACF® Users Training 2005
SECURITY RELOADED

Kerberos on z/OS
Three Heads Are Better Than One

Network Authentication Service
and
Resource Access Control Facility



+

Eric Rosenfeld
z/OS Security Development
rosenfel@us.ibm.com

May 2005

Agenda

- General Kerberos Overview
- Kerberos Registry Support Overview
- Getting Started
  - Server Information
  - Registry set-up
- SAF Callable Services
- Dependencies and Migration Considerations
- z/OS V1R4 and V1R6 extensions
- Session Summary

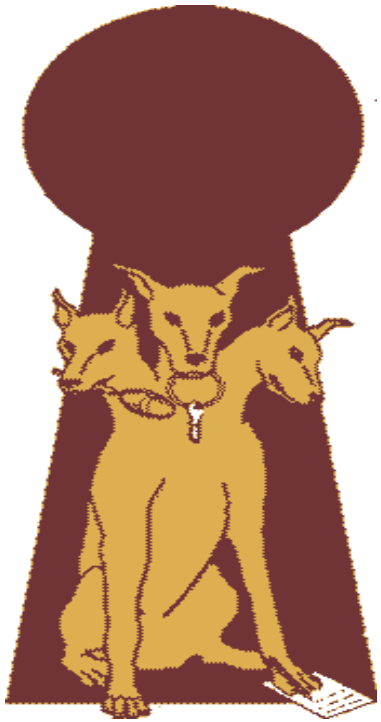
Trademarks

- The following are trademarks or registered trademarks of the International Business Machines Corporation:
  - IBM, DB2, OS/390, RACF, SecureWay, S/390
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Kerberos is a trademark of MIT
- Other company, product, and service names may be trademarks or service marks of others.

Greek Mythology

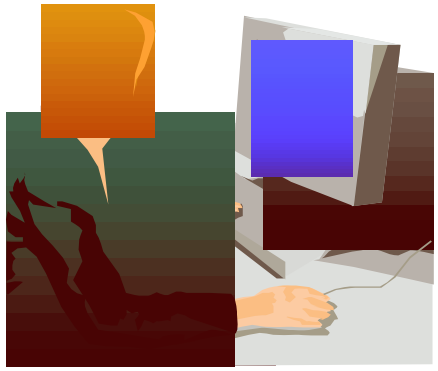
Kerberos (Cerberus) was the mythological three-headed dog that guarded the entrance to the underworld.

Unless you could get past Kerberos, you could not enter (or leave!) the underworld



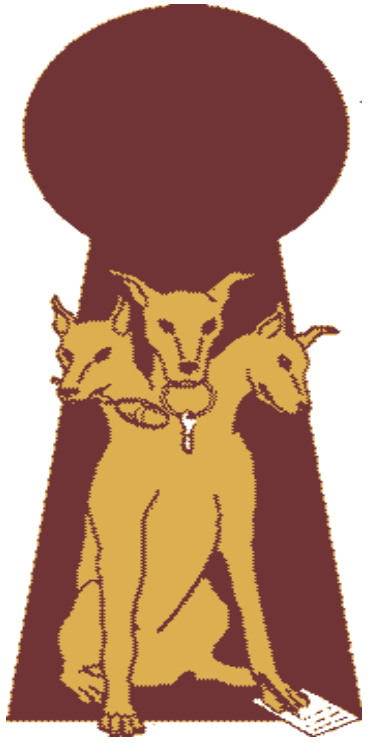
What is Kerberos?

- A distributed authentication service developed by MIT
- Allows user authentication over a physically untrusted network
- Tickets are issued by a Kerberos authentication server
  - Users and servers are required to have keys registered with server
- Flows to and from server establish a session key
  - used in a direct exchange between a user and a service
- V5 implemented in OS/390, z/OS, AIX, AS/400, Win2K, Solaris
  - **Network Authentication Service** component of Integrated Security Services on z/OS



Client

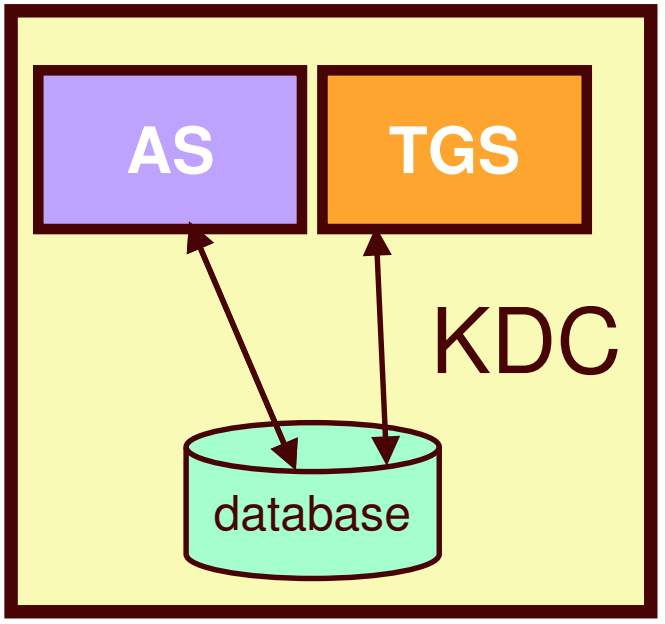
Trusted
Third
Party



Server

Key Distribution Center (KDC)

- Trusted "third party“
  - Both client and server trust the information in/decisions of the KDC
- Responsible for issuing user credentials and tickets
- Consists of
  - an authentication server (KAS)
    - Authenticates users
    - Grants Ticket Granting Tickets
  - a ticket granting server (TGS)
    - Generates session key
    - Grants service tickets
  - a Kerberos Data Base (KDB)
    - Contains keys for each user and server



Additional Terms

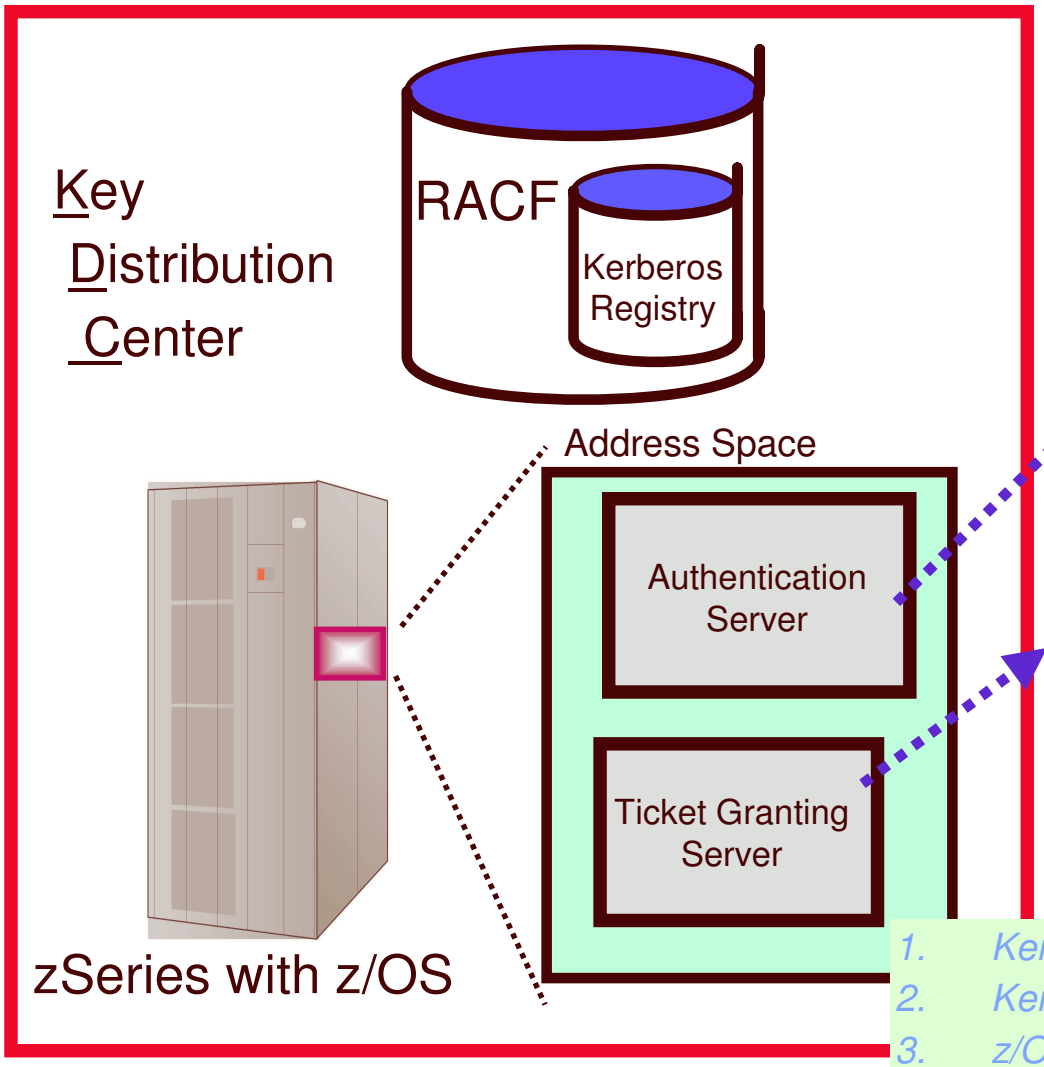
- Ticket
  - An encrypted electronic authentication token including:
    - client's identity
    - a dynamically created session key
    - a time stamp
    - lifetime for the ticket
    - a service name
- Realm
  - The Kerberos domain: the set of entities which authenticate using the domain of authority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

Ticket Use

- At logon (kinit) Ticket Granting Ticket returned
- To use a service, TGT presented w/request
- Server returns service ticket
  - Contains session key
  - Client presents service ticket to server as part of authentication protocol
    - GSS-API gss_init_sec_context method
  - Can be used until expiration
  - Avoids repeated authentication

Kerberos on z/OS and OS/390

(Its own component, integrated with RACF via SAF)



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

(AS)

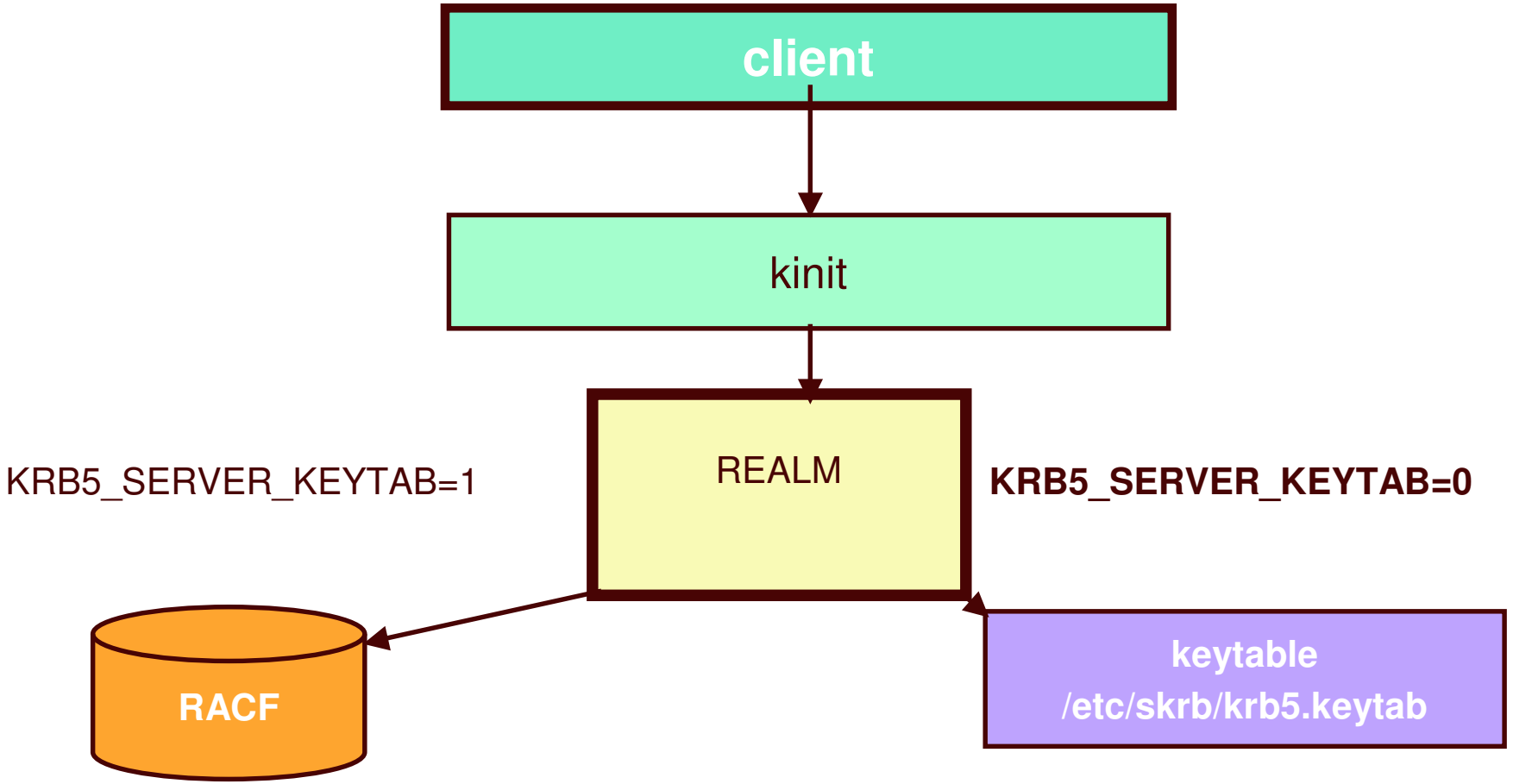
- Authenticates Users
- Grants TGTs

(TGS)

- Generates Session Keys
- Grants service tickets based on TGT

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

Network Authentication Service - keytab or RACF

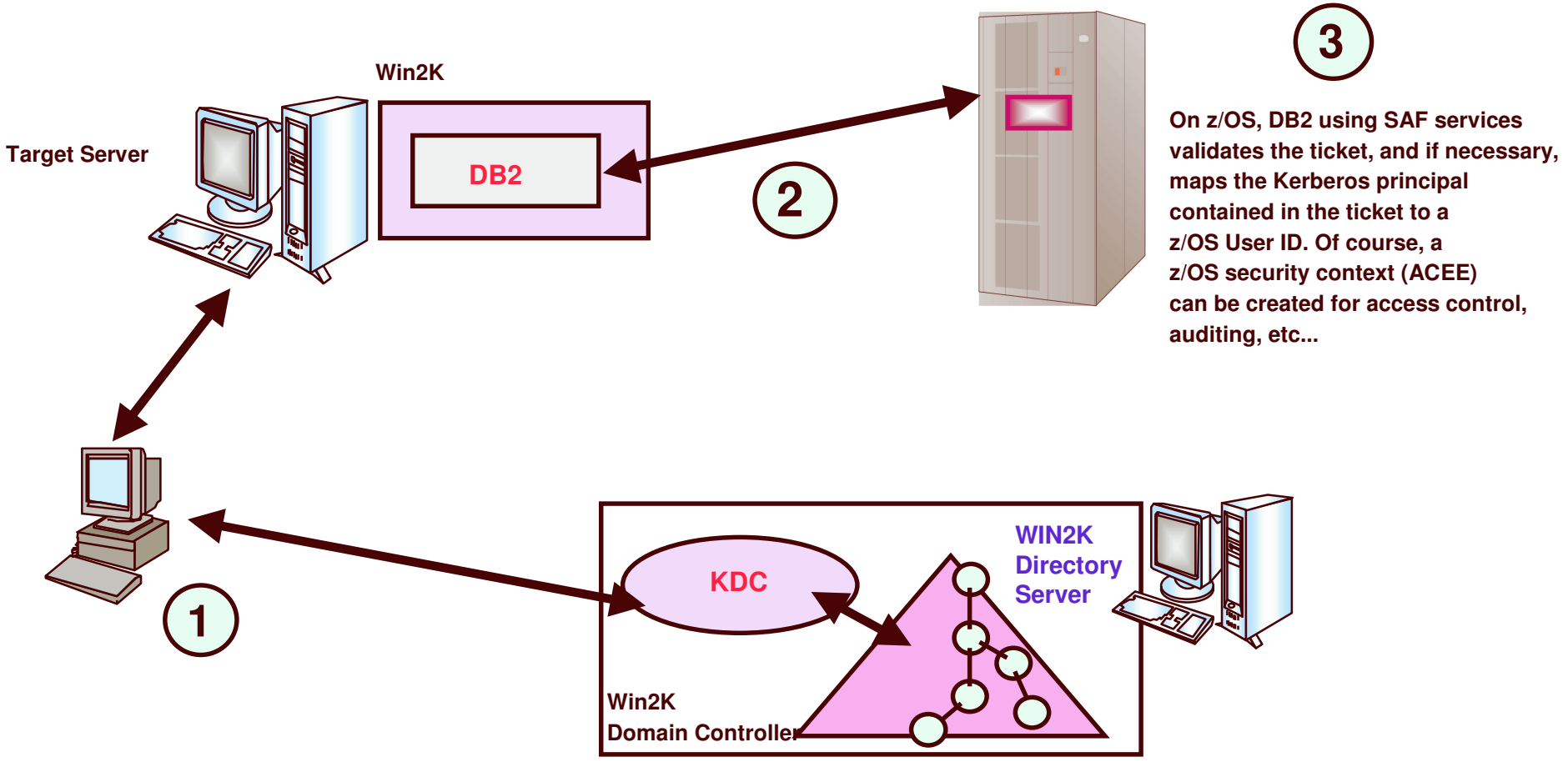


User/application needs read access to IRR.RUSERMAP

z/OS and WIN2K Kerberos Domain

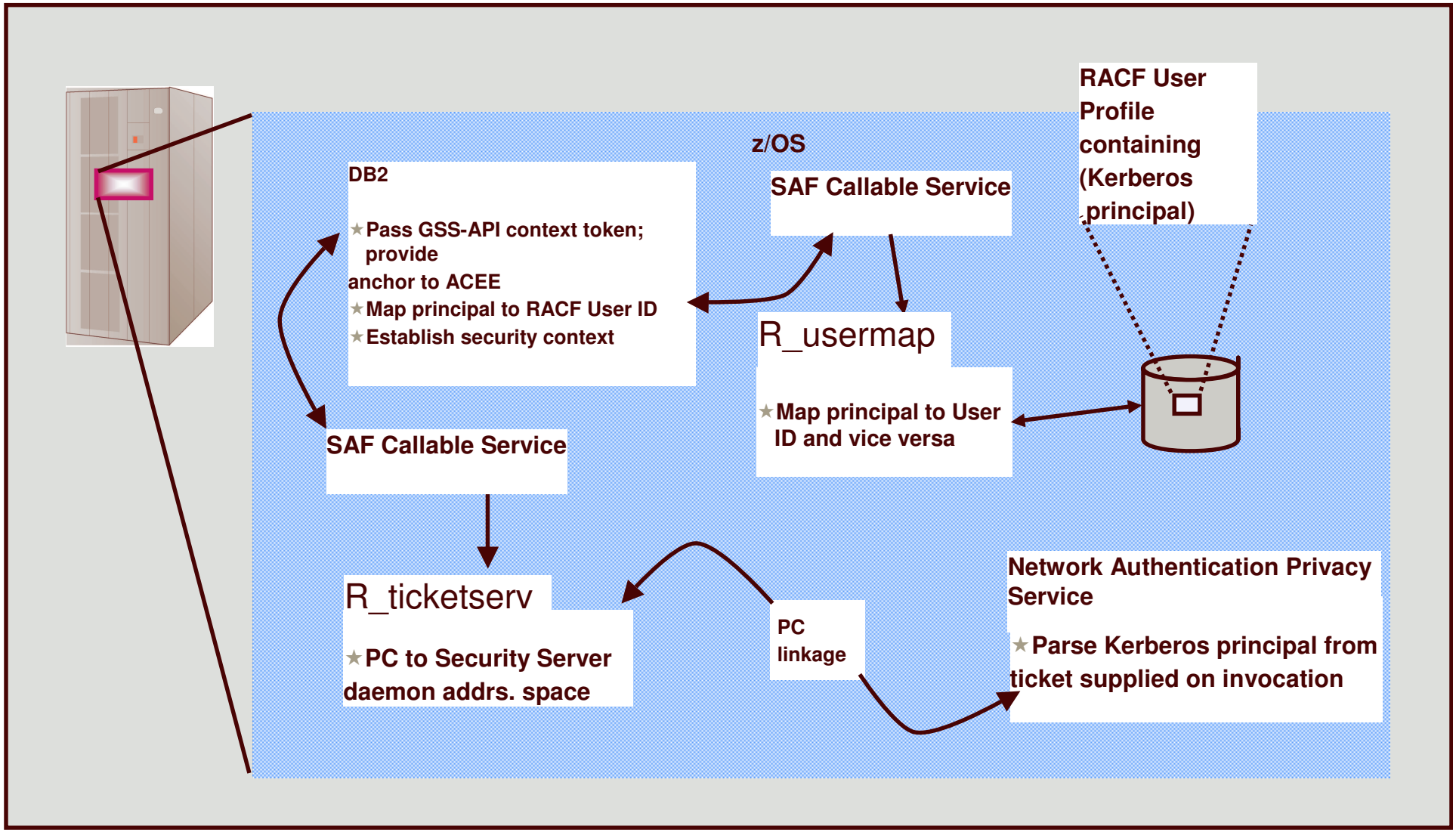
The client authenticates to the KDC, and obtains a ticket for the target server.

The assumption in this chart, is that the target server is Win2k running DB2, and the target server makes a request to a DB2 instance on z/OS. The DB2 instance on the target server passes the ticket of the user client on the flow to the z/OS host.



z/OS and WIN2K Kerberos Domains...

This pictorial indicates that z/OS needs to be a viewed as a Kerberos peer domain. Administratively, a peer trust relationship has been established between the z/OS Kerberos domain and a Win2K Kerberos domain. Local Kerberos principals must be defined to the z/OS 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 z/OS Kerberos domain.



Network Authentication Service – Commands

- kinit - obtains or renews the Kerberos ticket-granting ticket.
- klist - displays the contents of a Kerberos credentials cache or key table.
- kdestroy - destroys a Kerberos credentials cache.
- keytab - manages a key table (z/OS likely will use RACF).
- ksetup - manages Kerberos service entries in the LDAP directory for a Kerberos realm.
- kpasswd - allows principal to change password
- kvno - returns key version number.
- kadmin - administer non z/OS KDC with Kerberos commands
  - help, list_principals, add_principal, delete_principal, change_password, rename_principal, list_policies, add_policy, delete_policy, add_key, etc.

Network Authentication Service – Console Commands

- DISPLAY
  - creds, owner, date - contents of credentals cache DB
  - XCF - active security servers in SYSPLEX
  - CRYPTO - list of available encryption types, hardware crypto availability and whether crypto may be used on user data
  - LEVEL – service level of server

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 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 z/OS 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
    - A TGT issued in one realm can be used in another

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 1777
  - System datasets
    - Add EUVF.SEUVFLPA to LPALST
    - Add EUVF.SEUVFLNK to LNKLST
    - Add EUVF.SEUVFEXC to SYSEXEC DD concatenation for TSO

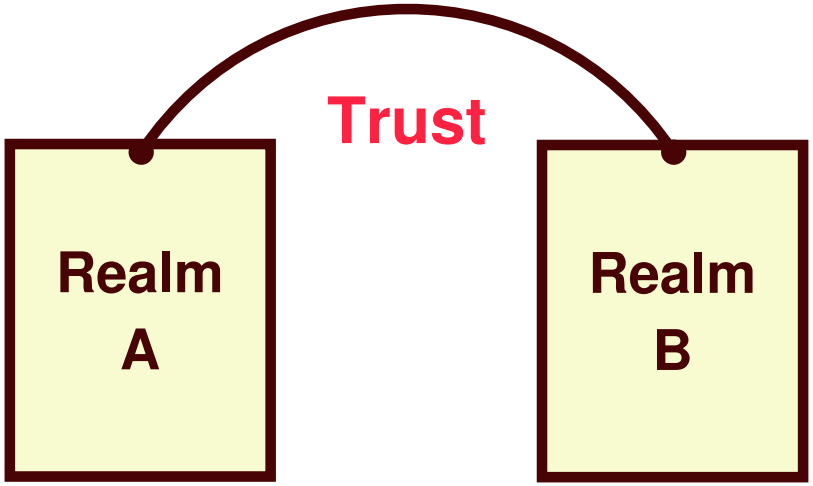
Network Authentication Service - Installation

- 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
- **ENCRYPT** – Supported encryption types
  - Choice of DES, Triple DES and DES with Derivation
- **NODEFTKTLFE, NOMAXTKTLFE, NOKERBNAME, NOMINTKTLFE, NOPASSWORD**, **NOENCRYPT** 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
  - **ENCRYPT** specifies supported encryption types
    - Choice of DES, Triple DES and DES with Derivation
  - **NOMAXTKTLFE, NOKERBNAME, NOENCRYPT, 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
- SETR KERBLVL determines what level of encryption can be supported (Default – 1)
  1. DES
  2. DES, Triple DES, DES w/Derivation

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

- 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
  - z/OS V1R2 and above requires Kerberos sever be installed prior to any key generation
  - KERBLVL SETROPTS setting should not be lowered

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 Telnet Server
z/OS V1R2 RSH Server
z/OS V1R2 LDAP
z/OS V1R5 EIM

z/OS V1R4

z/OS R4 Updates

- TCP/IP V6 supported
- NDBM (New DataBase Manager) support
  - UNIX backed SAF database alternative
  - Not shared by SYSPLEX
  - SAF still required to map principals to RACF IDs
  - kadmin used for administration

z/OS V1R6

z/OS R6 Updates

- Network Time Offset support
  - Allows setting of offset from real time to allow for system not having “real” time
- SYSPLEX credential cache can run w/o starting a KDC on that image
- Exploits CPACF hardware on T-Rex machines
- New library for executables
  - Moved from EUVF.EUVFLNK to SYS1.SIEALNKE

z/OS R6 Updates (SAF/RACF)

- SAF Callable Service GSS-API support
  - Allows Kerberos GSS-API function via non-LE interface
  - R_GenSec service provides following GSS-API functions:

1. GSEC_INIT_SEC_CONTEXT
2. GSEC_CONT_SEC_CONTEXT
3. GSEC_ACC_SEC_CONTEXT
4. GSEC_DEL_SEC_CONTEXT
5. GSEC_REL_CRED
6. GSEC_GET_MIC
7. GSEC_VER_MIC
8. GSEC_WRAP_MSG
9. GSEC_UNWRAP_MSG
10. GSEC_EXPORT_SEC_CONTEXT
11. GSEC_IMPORT_SEC_CONTEXT
12. GSEC_EXPORT_CRED
13. GSEC_IMPORT_CRED
14. GSEC_ACQUIRE_CRED

Session Summary

- What we have covered:
  - What Kerberos is and does
  - How SAF/RACF interacts with the Network Authentication Service
  - How an application would interact with SAF to map Kerberos constructs to RACF constructs
  - How to install and configure Kerberos support
  - An overview of newer support

References

IBM Books

- SA22-7691 z/OS Security Server RACF Callable Services
- SA22-7687 z/OS Security Server RACF Command Language Reference
- GA22-7680 z/OS Security Server RACF Data Areas
- SA22-7682 z/OS Security Server RACF Macros and Interfaces
- SA22-7686 z/OS Security Server RACF Messages and Codes
- SA22-7683 z/OS Security Server RACF Security Administrator's Guide
- SC24-5926 z/OS Integrated Security Services Network Authentication and Privacy Service Administration
- SC24-5927 z/OS Integrated Security Services 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

Internet

- http://web.mit.edu/kerberos/www/

Questions ?