

New York and Tampa RUG

Kerberos on z/OS

Network Authentication Service
and
Resource Access Control Facility



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

February 12, 2008

IBM Systems and Technology Group




Agenda

- General Kerberos Overview
- Base Kerberos Registry Support Overview
- Getting Started
 - ▶ Server Information
 - ▶ Registry set-up
- SAF Callable Services
- Dependencies and Considerations
- Session Summary

© 2008 IBM Corporation


2

IBM Systems and Technology Group 

Trademarks

- The following are trademarks or registered trademarks of the International Business Machines Corporation:
 - ▶ IBM, DB2, OS/390, RACF, SecureWay, z/OS, AS/400, AIX
- 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.
- SOLARIS is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries
- Kerberos is a trademark of MIT
- Other company, product, and service names may be trademarks or service marks of others.


© 2008 IBM Corporation 3

IBM Systems and Technology Group 

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



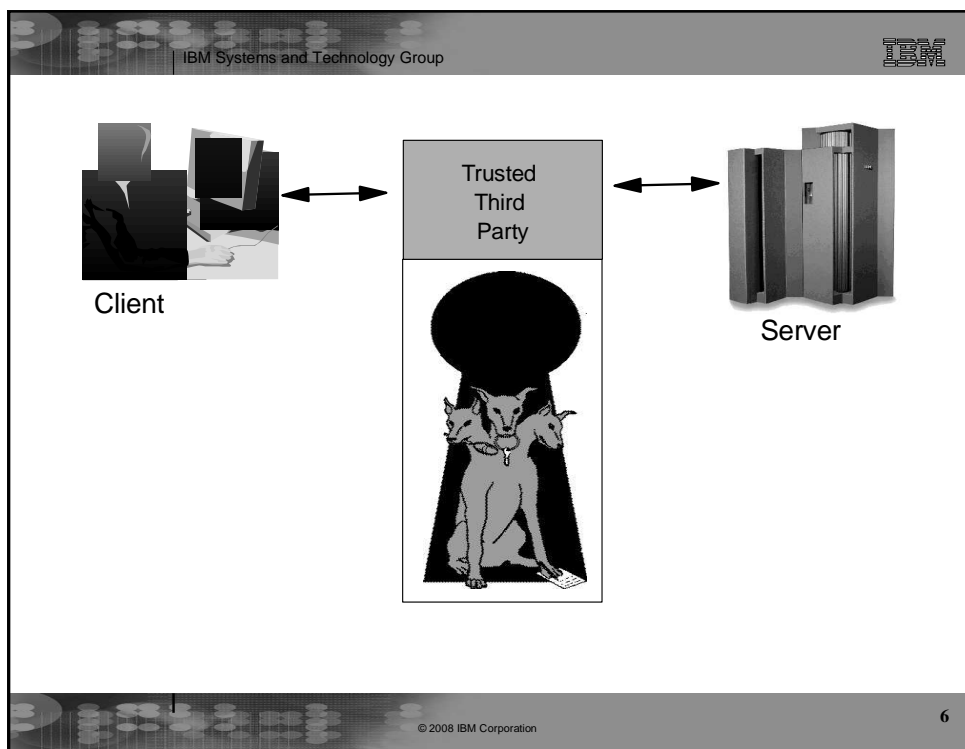
© 2008 IBM Corporation 4

IBM Systems and Technology Group

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 covered by a session key
 - used in a direct exchange between a user and a service
- V5 implemented in OS/390, z/OS, AIX, AS/400, Windows, Solaris and others
 - **Network Authentication Service** component of Integrated Security Services on z/OS

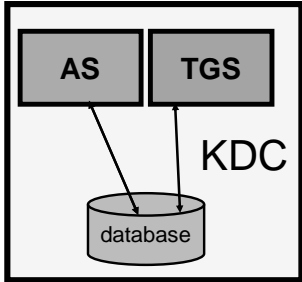
© 2008 IBM Corporation 5



IBM Systems and Technology Group

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



© 2008 IBM Corporation

7

IBM Systems and Technology Group

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

© 2008 IBM Corporation

8

IBM Systems and Technology Group

IBM

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

9

© 2008 IBM Corporation

IBM Systems and Technology Group

IBM

Kerberos on z/OS

(Its own component, integrated with RACF via SAF)

The diagram shows a zSeries with z/OS connected to a Key Distribution Center (KDC). The KDC is integrated with RACF and contains a Kerberos Registry. The KDC's address space contains an Authentication Server and a Ticket Granting Server.

Standards

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

(AS)

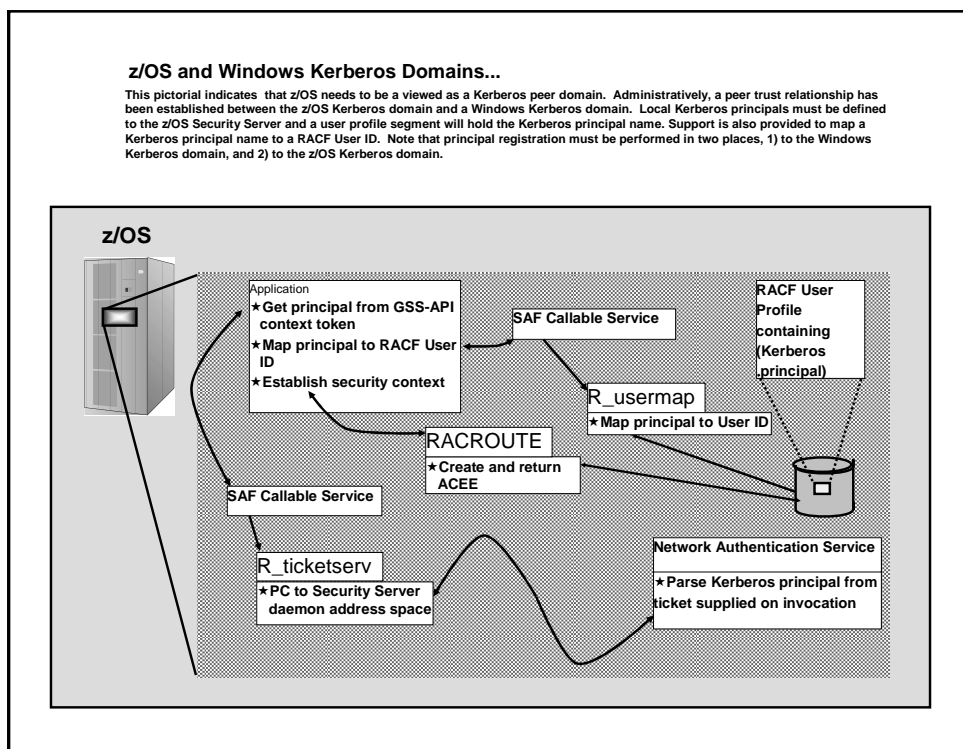
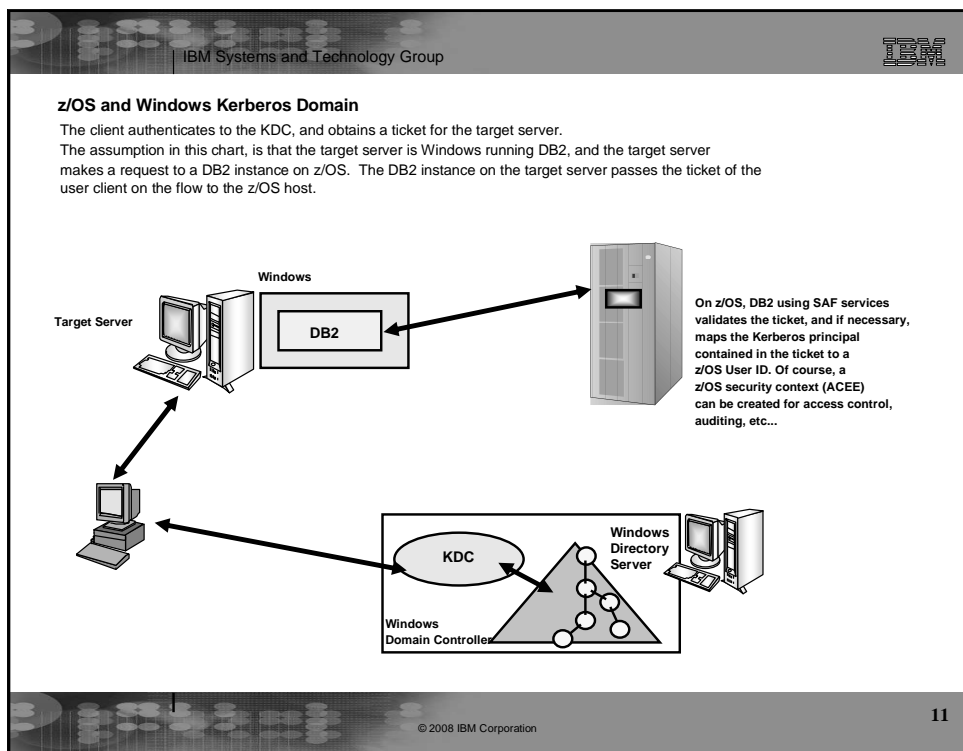
- Authenticates Users
- Grants TGTs


(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

© 2008 IBM Corporation




IBM Systems and Technology Group 

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.


© 2008 IBM Corporation 13

IBM Systems and Technology Group 

RACF is the Kerberos Registry

- The 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


© 2008 IBM Corporation 14

IBM Systems and Technology Group 

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


© 2008 IBM Corporation 15

IBM Systems and Technology Group 

Kerberos Registry

- ▶ 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

© 2008 IBM Corporation 16


IBM Systems and Technology Group 

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

17

© 2008 IBM Corporation

IBM Systems and Technology Group 

SAF 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_ticketerv is called by applications to determine the principal name associated with a credential
- R_usermap is called by applications to map from principal to RACF identifier

18

© 2008 IBM Corporation

IBM Systems and Technology Group

SAF Services (cont)

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_EXPORT_CRED
 12. GSEC_IMPORT_SEC_CONTEXT
 13. GSEC_IMPORT_CRED
 14. GSEC_ACQUIRE_CRED


© 2008 IBM Corporation 19

IBM Systems and Technology Group

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


© 2008 IBM Corporation 20

IBM Systems and Technology Group 

Network Authentication Service - Installation

- Installs into
 - ▶ UNIX file system
 - executables in directory /usr/lpp/skrb
 - /etc/skrb files need access 755
 - /var/skrb/creds needs access 1777
 - ▶ System datasets
 - EUVF.SEUVFLPA
 - SYS1.SIEALNKE (EUVF.SEUVFLNK Pre V1R6)
 - EUVF.SEUVFEXC for SYSEXC DD concatenation for TSO

© 2008 IBM Corporation 21

IBM Systems and Technology Group 

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

© 2008 IBM Corporation 22

IBM Systems and Technology Group

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

© 2008 IBM Corporation 23

IBM Systems and Technology Group

Registry Definitions

```
graph LR; A[Realm A] --- Trust --- B[Realm B]
```

Commands must be entered to define:

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

© 2008 IBM Corporation 24

IBM Systems and Technology Group

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

© 2008 IBM Corporation 25

IBM Systems and Technology Group

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

© 2008 IBM Corporation 26

IBM Systems and Technology Group

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`


© 2008 IBM Corporation 27

IBM Systems and Technology Group

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(passwr1))`
 - ▶ `RDEFINE REALM ../../KRB2000.IBM.COM/krbtgt/KRB390.IBM.COM KERB(PASSWORD(passwr2))`

© 2008 IBM Corporation 28


IBM Systems and Technology Group 

User Commands

- Local principal definition with ADDUSER/ALTUSER
 - Local realm must exist before issuing command
 - **MAXTKLFE** 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
 - **NOMAXTKLFE, NOKERBNAME, NOENCRYPT, NOKERB** only valid on ALTUSER
 - Kerberos keys generated at non-expired password setting
 - KERBLINK mapping profile created/updated

29

© 2008 IBM Corporation

IBM Systems and Technology Group 

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 ← key
```

30


© 2008 IBM Corporation

IBM Systems and Technology Group


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

© 2008 IBM Corporation
31


IBM Systems and Technology Group


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

© 2008 IBM Corporation
32

IBM Systems and Technology Group




z/OS Tid Bits

- 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

© 2008 IBM Corporation

33

IBM Systems and Technology Group



Dependencies and Gotchas


- Network Authentication Service implements V5 standard
- Any application can use R_ticketerv and R_usermap to map Kerberos information to RACF
- Kerberos sever required to be installed prior to any key generation
- RRSF local node must be defined to allow for keys to be generated for user password application updates
- Password must be changed after user definition to generate initial keys

© 2008 IBM Corporation

34



-
- IBM Systems and Technology Group
- IBM
- **The z/OS Network Authentication Service has been enhanced to support the AES cryptographic algorithm**
 - AES support promotes interoperability with other non-z/OS Kerberos implementations
 - RACF can act as the registry for the z/OS Network Authentication Service, so RACF management interfaces are also being extended for z/OS Kerberos AES support.
 - **Implementation of SPKM-3/LIPKEY standards**
 - Positions the z/OS Network Authentication Service to be able to interoperate with other non-z/OS GSS-API implementations using existing certificate infrastructure
- © 2008 IBM Corporation
- 36

IBM Systems and Technology Group 

RACF interface changes for AES

- **Commands, panels, utilities, and SAF callable services which support Kerberos encryption types are enhanced to also support 128-bit and 256-bit AES.**

Allowed on both USER and REALM class profiles


```
ADDUSER RONTOMS KERB(KERBNAME(raeburn) ENCRYPT(NOAES256))

LISTUSER RONTOMS NORACF KERB
USER=RONTOMS

KERB INFORMATION
-----
KERBNAME= raeburn
KEY ENCRYPTION TYPE= DES DES3 DESD AES128 NOAES256
```

- **Note that using a command or panel to enable use of AES keys, does not generate new keys...a password change is also required!**

© 2008 IBM Corporation 37

IBM Systems and Technology Group 

The GSS-API

- **Generic Security Service Application Programming Interface (GSS-API) support is provided by the z/OS Network Authentication Service**
 - The GSS-API is a set of programming interfaces which abstract identity authentication, message origin authentication and integrity, and message confidentiality
 - In concept, a secure application developed using the GSS-API should be able to work over different security mechanisms without changes to the application
- **Previously, the z/OS Network Authentication Service GSS-API offering only supports the Kerberos security mechanism**
- **LIPKEY and SPKM-3 support has been added as extensions to the GSS-API**

© 2008 IBM Corporation 38

IBM Systems and Technology Group

IBM

SPKM-3

- **The Simple Public-Key GSS-API Mechanism (SPKM) is based on a public key infrastructure, not the Kerberos symmetric-key infrastructure**
 - SPKM-3 does not use secure timestamps, enabling secure authentication in environments without access to secure time
 - Designed to be flexible, for example providing Algorithm Identifiers for specifying various algorithms to be used by communicating peers
 - Provides support for asymmetric algorithm-based digital signatures
 - Data formats and procedures are designed to be as similar to the Kerberos mechanism as possible for ease of implementation by applications which are already Kerberos enabled
- **SPKM-3 uses the same certificate infrastructure as SSL**

© 2008 IBM Corporation

39

IBM Systems and Technology Group


IBM

LIPKEY

- **LIPKEY (a Low Infrastructure Public Key Mechanism using SPKM) is a GSS-API security mechanism which can be used when the initiator (client) does not have a certificate and instead uses user ID and password for authentication**
- **It consists of a client with no public key certificate, accessing a server with a public key certificate (in contrast, in SPKM-3, both client and server require access to certificates)**
- **The server must have access to a user ID/password repository (we use the __passwd system routine, with setup/restrictions documented in the z/OS Network Authentication Service Programming Guide)**


© 2008 IBM Corporation

40


IBM Systems and Technology Group 

How LIPKEY works

A client using the LIPKEY mechanism

- Obtains the server's certificate
- Verifies that it was signed by a trusted CA 
- Generates a random session symmetric key
- Encrypts the session key with the server's public key
- Sends the encrypted session key to the server
- At this point, the client and server have a secure channel, so the client can provide a user name and password for authentication

© 2008 IBM Corporation 41


IBM Systems and Technology Group 

What externals were changed?

- New z/OS Network Authentication Service environment variables are added, such as `GSS_KEYRING_NAME` (specifies the name of the key database HFS file or the SAF key ring)
- New messages are added
- GSS-API descriptions, parameter descriptions, and parameter format descriptions are modified to indicate/provide support for the two new security mechanisms, SPKM-3 and LIPKEY

For example, the `desired_mech` parameter of the `gss_acquire_cred` function is modified to indicate that `gss_mech_spkm3` and `gss_mech_lipkey` are now supported in addition to `gss_mech_krb5`


© 2008 IBM Corporation 42

IBM Systems and Technology Group 

Migration & Coexistence Considerations

- Problems can occur when RACF is the Kerberos registry and the database is shared between z/OS V1R9 and lower-level systems
 - As always, administration should be done on the higher level system
 - The fix for RACF APAR OA20304 (V1R7 PTF UA33765 / V1R8 PTF UA33766) must be applied in order for Kerberos to use **triple DES** and **DES with derivation** correctly on the lower-level systems

© 2008 IBM Corporation 43

IBM Systems and Technology Group 

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

© 2008 IBM Corporation 44

IBM Systems and Technology Group

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/>

© 2008 IBM Corporation 45

IBM Systems and Technology Group

References for V1R9 Enhancements

- RFC archives
 - RFC 2025 - The Simple Public-Key GSS-API Mechanism (SPKM)
 - RFC 2847 - LIPKEY - A low infrastructure mechanism Using SPKM
 - RFC 3962 - Advanced Encryption Standard (AES) Encryption for Kerberos
 - RFC 4121 - The Kerberos V5 GSSAPI Mechanism: Version 2
 - RFC2253 UTF-8 String Representation of Distinguished names
 - RFC2459 X.509 Public Key Infrastructure
- SC24-5926 z/OS Network Authentication Service Administration
- SC24-5927 z/OS Network Authentication Service Programming
- SC24-5901 Cryptographic Services System Secure Sockets Layer Programming
- GA22-7800 z/OS Unix System Services Planning
- SA22-7803 z/OS Unix System Services Programming: Assembler Callable Services Reference

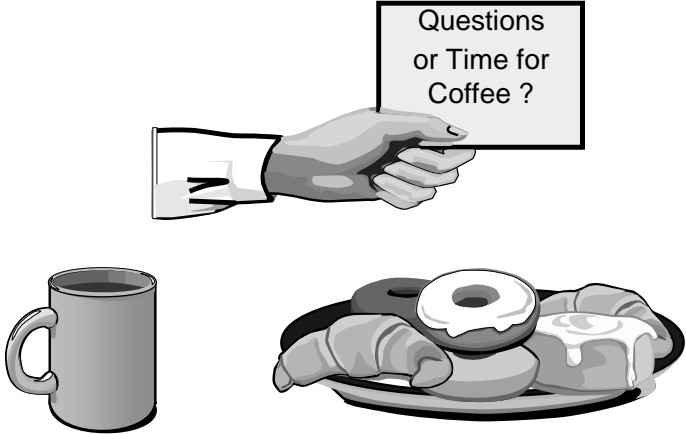
© 2008 IBM Corporation 46

IBM Systems and Technology Group

IBM

Questions ?

Questions
or Time for
Coffee ?



© 2008 IBM Corporation

47

This slide features a header with the IBM logo and the text 'IBM Systems and Technology Group'. The main content area has a large title 'Questions ?' and a smaller box containing the text 'Questions or Time for Coffee ?'. Below this, there is an illustration of a hand in a white sleeve holding the sign. To the left of the hand is a coffee mug, and to the right is a plate with a croissant, a donut, and a slice of cake. The footer contains the copyright notice '© 2008 IBM Corporation' and the slide number '47'.

IBM Systems and Technology Group


IBM

Reference

© 2008 IBM Corporation

48


This slide features a header with the IBM logo and the text 'IBM Systems and Technology Group'. The main content area has a large, stylized button with the word 'Reference' written on it. The button has a 3D effect with a shadow. The footer contains the copyright notice '© 2008 IBM Corporation' and the slide number '48'.

IBM Systems and Technology Group 

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

© 2008 IBM Corporation 49

IBM Systems and Technology Group 

R_usermap (IRRSIM00)

- Map application user
 - ▶ Function codes:
 - 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

© 2008 IBM Corporation 50

IBM Systems and Technology Group

R_admin (IRRSEQ00)

- **Functions supported**
 - ADMN_ADD_USER, ADMN_ALT_USER, ADMN_LST_USER
ADMN_ADD_GENRES, ADMN_ALT_GENRES,
ADMN_LST_GENRES to support KERB segment fields
- **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

© 2008 IBM Corporation

51