Session 02

z/OS V1R9
Security Server Update

Prepared by Patrick Kappeler
IBM Consulting IT Specialist
kappeler@fr.ibm.com

Redbooks
International Technical Support Organization

z Security Update

Trademarks

See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

- Password Phrase Minimum Length
- Java APIs For z/OS Security

Password Phrase Minimum Length

RACF Password Extensions

Background

- Longer passwords desired by customers but the RACF password length (8 characters) cannot be modified due to very deep integration in the operating system and APIs
- The z/OS two-step approach
  - 8-character mixed-case passwords at z/OS V1R7
  - Password Phrase (a.k.a. Passphrase) support at z/OS V1R8
    - Character string, 14 to 100 characters in length
    - Requires changes in applications which currently support passwords and want to support phrases (new keywords when calling SAF)
    - Users can have both a password and password phrase at the same time. It is expected that this will be common for some time

The first Password Phrase exploiter is HCM at z/OS V1R9

RACF Password Extensions

Background

The Password Phrase has fixed syntax rules:

•The user ID (as sequential upper case characters or sequential lower case characters) is not part of the password phrase

•At least 2 alphabetic characters are specified (A - Z, a - z)

•At least 2 non-alphabetic characters are specified (numeric characters, punctuation, special characters)

•No more than 2 consecutive characters are identical

A new Password-Phrase exit (ICHPWX11) can be used to install customized syntax rules

RACF Password Phrase – z/OS V1R9 Improvements

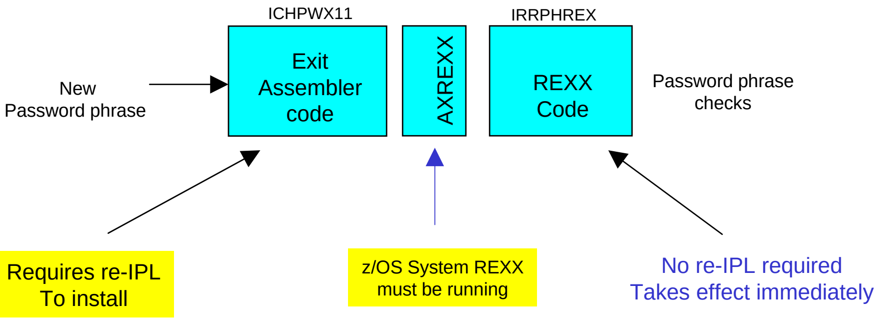
1. Allow a password phrase minimum length of 9 characters (instead of 14) – Maximum is still 100 chars
   - Password phrases of length 9-13 characters in length may be specified if
     - the installation has coded the ICHPWX11 password phrase quality exit
     - and the exit accepts the shorter password phrase.
   - If the ICHPWX11 password phrase exit is not present, the minimum password phrase length remains 14.
2. Provides a REXX sample password phrase quality rules in REXX
   - Robust, easy to code, easy to change, and immediately effective password phrase quality rules
   - Exploits the z/OS System REXX facility
3. Make the coding of authentication routine, using RACROUTE REQUEST=VERIFY/X more easy
   - The RACROUTE VERIFY process can automatically recognize password (length < 9 chars) and password phrase (length >= 9 chars)

No support yet for password phrase enveloping

RACF Password Phrase – z/OS V1R9 Improvements

Coding password phrase quality rules in REXX

- A sample assembler code for the ICHPWX11 exit
  - source code in SYS1.SAMPLIB(RACEXITS)
  - accumulates a number of parameters and then passes them to IRRPHREX, using the new z/OS System REXX facility



Requires re-IPL To install

z/OS System REXX must be running

No re-IPL required
Takes effect immediately

- The sample SYS1.SAMPLIB(IRRPHREX) REXX exec Implements check for:
  - Maximum/minimum length
  - Allowable characters
  - Leading/trailing blanks
  - User name allowed or not
  - Triviality checks with respect to previous phrase
  - Minimum unique characters/words with respect to previous phrase
  - Dictionary check

Java APIs
For z/OS
Security Services

z/OS Security Services – Java APIs

APIs provided in z/OS

- RACF Passticket Java evaluation and generation (z/OS V1R7)
  /usr/include/java_classes/IRRRacf.jar & IRRRacfDoc.jar
- EIM Java client (z/OS V1R7)
  /usr/lpp/eim/lib/
- RACF users and groups administration – JSec

New at z/OS V1R9

APIs provided in the IBM SDK for z/OS

SAF classes (JDK V1.R4): PlatformAccessControl, PlatformThread, PlatformSecurityServer, PlatformAccessLevel, PlatformReturned, PlatformUser

See the appendix for further details

Java And z/OS Security Server

New at z/OS V1R9

Java Interface To Users And Groups (JSec)

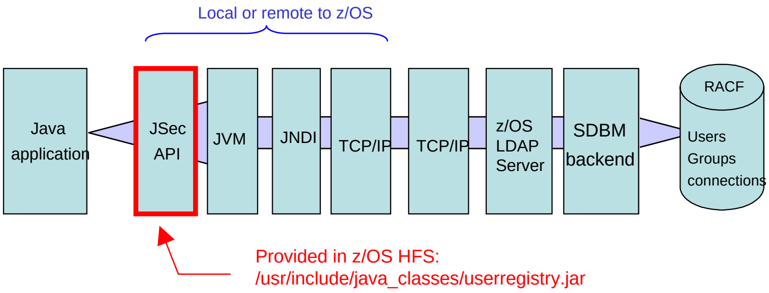
- Two parts
  - Generic interface (true Java interface) that could be used to query users and groups in other security repositories.
  - RACF implementation of this interface – Maps the ADDUSER, ALTUSER, CONNECT, …, commands
- Extensible - to allow for future RACF enhancements or use by other security repositories.
- Built on commonly used objects and interfaces in JSDK
  e.g. javax.naming.directory.BasicAttributes, javax.naming.directory.ModificationItem
- Can be run ON or OFF z/OS platform (LDAP interface used)
  Access RACF through the SDBM LDAP backend

www-03.ibm.com/servers/eserver/zseries/software/java/jsec/overview.html

Java And z/OS Security Server

New at z/OS V1R9

Java Interface To Users And Groups (JSec) Implementation



Provided in z/OS HFS:
/usr/include/java_classes/userregistry.jar
/usr/include/java_classes/RACFuserregistry.jar

```
import com.ibm.eserver.zos.racf.userregistry.*;
import com.ibm.security.userregistry.*;
import javax.naming.*;
import javax.naming.directory.*;

public class sample {
public static void main(String[] args)
   {
   RACF_remote remote = new
RACF_remote("ldap://alps4214.pok.ibm.com:389",
      "simple",
      "IBMUSER",
      "secret", "o=racfdb,c=us");
try
   {
     SecAdmin racfAdmin = new RACF_SecAdmin(remote);
     if (racfAdmin != null)
       {
       User ibmuser = racfAdmin.getUser("ibmuser");
       BasicAttributes ibmuser_attr = ibmuser.getAttributes();
       System.out.println("Attributes returned for IBMUSER are: ");
       RACF_SecAdmin.displayAttributes(ibmuser_attr);
       }
```

```
Attributes returned for IBMUSER are:
BASE_CREATED: 11/04/94
BASE_DAYS: SUNDAY, MONDAY, TUESDAY,
        WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
BASE_DFLTGRP: SYS1
BASE_LAST-ACCESS: 06/09/06/17:32:24
BASE_OPERATIONS: No values ← boolean attribute
BASE_OWNER: IBMUSER
BASE_PASS-INTERVAL: 30
BASE_PASSDATE: 06/05/06
BASE_PASSWORD: Password Exists
BASE_SECLABEL: SYSMULTI
BASE_SPECIAL: No values ← boolean attribute
BASE_TIME: ANYTIME
BASE_USERID: IBMUSER
OMVS_PROGRAM: /bin/sh
OMVS_UID: 0
```

Thank You

Any Questions ?

Appendix

- See the following RACF books for more details
  - Security Server RACF System Programmer's Guide (SA22-7681)
  - Security Server RACF Security Administrator's Guide (SA22-7683)
  - Security Server RACF Command Language Reference (SA22-7687)
  - OSecurity Server RACROUTE Macro Reference (SA22-7692)

- And the following z/OS books for System REXX
  - MVS Assembler Services Guide (SA22-7605)
  - MVS Programming: Authorized Assembler Services Reference, Volume 1 (ALESERV-DYNALLOC) (SA22-7609)
  - MVS System Commands (SA22-7627)
  - MVS Initialization and Tuning Reference (SA22-7592)

z/OS SAF Interfaces

- Java static class methods provide an interface to the z/OS Security Server using SAF (System Authorization Facility) and z/OS services to provide basic authentication and authorization services.
  - PlatformSecurityServer class
    - IsActive(), resourceIsActive()
  - PlatformUser class
    - authenticate(), changePassword(), isUserInGroup()
  - PlatformAccessControl.checkPermission()
  - PlatformThread.getUserName()
- z/OS documentation available at http://www.ibm.com/servers/eserver/zseries/software/java/security14.html

Java SAF classes (JDK V1R4 and above)

These APIs are implemented through Java classes wrapping z/OS UNIX Services. The z/OS UNIX Services are in turn handled by a Security Server for z/OS that implements SAF interfaces (such as RACF).

The classes provided are:

- PlatformAccessControl
- PlatformThread
- PlatformSecurityServer
- PlatformAccessLevel
- PlatformReturned
- PlatformUser

These methods of these new classes allow a Java application to:

- Check to see if the Security Server or a specific security server class is active
- Extract the **userid** in effect for the current running thread
- Check the **userid** in effect for access rights to a resource
- Authenticate a **userid** and password

http://www-03.ibm.com/servers/eserver/zseries/software/java/j5security.html

RACF Passticket Java evaluation and generation (z/OS V1R7 and above)

- Java applications may now use the new IRRPassTicket class to generate and evaluate RACF PassTickets.

- The IRRPassTicket class is found in /usr/include/java_classes/IRRRacf.jar.
- IRRPassTicket uses native methods (JNI) to call r_tickerserv and/or r_gensec to perform PassTicket operations.

- JavaDoc documentation for the IRRPassTicket is located in /usr/include/java_classes/IRRRacfDoc.jar, which must be copied to a workstation, uncompressed and viewed with a web browser.

EIM Java client (z/OS V1R7 and above)

- Registry names in RACF profiles
  - –Methods in the ConfigurationMgr class will retrieve the names of the local SAF registry, Kerberos registry, or x.509 registry from the IRR.PROXY.DEFAULTS profile in the facility class.
  - –The names can be used on calls to the lookup methods – findTarget, findTargetFromSource, getAssociations, and getAssociatedEids

- EIM Domain name and bind dn and password stored in RACF profiles
  - –LDAPBIND class profile name stored in the LDAPPROF field in the EIM segment of the USER profile
  - –IRR.EIM.DEFAULTS profile in the LDAPBIND class or IRR.PROXY.DEFAULTS profile in the FACILITY class
    - DOMAINDN field in the EIM segment
    - LDAPHOST, BINDDN, and BINDPW fields in the PROXY segment.