# RACF® z/OS® V1R11 Update

## RACF Users Group of New England

November 2009

Mark Nelson, CISSP®, CSSLP™
z/OS® Security Server (RACF) Design and Development
IBM Poughkeepsie
markan@us.ibm.com





## Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Other company, product, or service names may be trademarks or service marks of others.

# Agenda

## Remember V1R10!

- Custom Fields
- Password Phrase Exploitation
- More Granularity in Allowing Password Reset
- Enhanced RACF Health Checks

## Introducing V1R11R11

- Program Object Signature Verification
- Logon Statistics Suppression
- Identity Propagation
- R_admin extract for General Resource
- Change logging for General Resource
- Automatic assignment of UID and GID to users of Unix System Services
- Profile name in Authorization Exits
- IRRADU00 support for WAS and TKLM
- RACDCERT multi-byte character improvements
- PKI Private Key recovery
- PKI Web Pages
- PKI Support for SHA256 with RSA signature algorithm






## z/OS Version 1 Release 11

## z/OS V1R11: Program Object Signature Verification

- **Allows the signing of program objects and the verification of the signature of program objects when the objects are loaded into storage**
  - BINDER: Creates signatures by calling RACF when the SIGN option has been specified
  - RACF: Stores the information (certificates, keys, and options) necessary for the signature generation and validation, calculates the signatures, performs the validations, and logs the results.
  - LOADER: Calls RACF when program objects are loaded
- **You can sign your own code and vendors can sign theirs**






## z/OS V1R11: Program Object Signature Verification…

- **Why sign code?**
  - “Belts and suspenders” or “defense in depth”: This support is intended to be used in conjunction with existing security mechanisms .
  - Digitally signing code can help increase the reliability and security of the system by adding an additional layer of controls on executable programs running on the system.
    - Digitally signing code makes it possible to detect changes to programs due to tampering or corruption.
    - Requiring that certain code be signed makes it easier to enforce change control procedures and protect against accidental changes to program code libraries. This helps avoid errors such as accidently placing 'test' code on a 'production' system.

## z/OS V1R11: Program Object Signature Verification…

- **RACF profiles are used to control program signing:**
  - Key ring associated with the user performing the signing
    - Contains the information appropriate for program signing (private key, X.509 certificates (signing, CertAuth) which themselves must be appropriately signed
  - IRR.PROGRAM.SIGNING profile(s) in the FACILITY class
    - Used to associate the key ring owner, key ring name, and message digest algorithm used in the signature generation and validation process.






## z/OS V1R11: Program Object Signature Verification…

- **RACF profiles are used to control program verification:**
  - IRR.PROGRAM.SIGNATURE.VERIFICATION profile in the FACILITY class
    - Used to associate the key ring owner and key ring name of the key ring which contains the signature verification key ring
  - Profiles in the PROGRAM class
    - Contains information options that specify the actions to be taken during verification process:
      - SIGREQUIRED: Is a signature required for this program? (YES,NO)
      - FAILLOAD: Under what conditions should the load fail? (ANYBAD, BADSIGONLY, NEVER)
      - SIGAUDIT: What should be logged? (ALL, SUCCESS, ANYBAD, BADSIGONLY, NONE)

## z/OS V1R11: Program Object Signature Verification…

- **Considerations:**
  - Only program objects (which must reside in in PDSEs) can be signed and verified.
    - Code in PDS or z/OS Unix System Services file system, or non Program Object code cannot be signed and verified. However, z/OS UNIX programs can 'link' to signed executables in PDSEs.
  - If a signed program is zapped (executable code changed), its signature is no longer valid.
  - IBM ships portions of the System SSL product as signed code.
  - Support is new for z/OS R11 and has been rolled back to z/OS R10.
  - Any installation or software provider can use these services to sign their own code.
  - Program objects are not encrypted






## z/OS V1R11: Logon Statistics Suppression

- **Allows you to specify which applications should only record on the first system access of a day**
  - Why? Reduced I/O and lower the impact of serialization on the RACF dataset.
- **APPL profiles are used to specify which applications are taking advantage of logon statistics suppression**
  - Specify “RACF-INITSTATS(DAILY)” anywhere in the APPLDATA
  - APPL class must be active and RACLISTed

## z/OS V1R11: Identity Propagation

- **Prior to z/OS V1R11, clients using distributed server applications which used a common server or application identity for transaction executing on z/OS would not be able to pass the identity of the end user to z/OS for logging**
- **With z/OS V1R11, applications can pass the distributed identity information abut the end user (distinguished name and realm) into z/OS where it will be used for logging**
  - Exploited by CICS TS 4.1
- **The distributed identity can be mapped to a RACF identity at:**
  - the distributed application server (as is often done today) or
  - the execution point on z/OS, using the new RACMAP support

```
RACMAP  [ID(mapped-to-userID)]
        MAP
            USERDIDFILTER(NAME('distributed-identity-username-filter'))
            REGISTRY(NAME('distributed-identity-registryname'))
            [WITHLABEL('label-name')]
      | DELMAP[(LABEL('label-name'))]
      | LISTMAP[(LABEL('label-name'))]
```






## z/OS V1R11: R_admin Enhancements

- **R_admin can now be used to extract information about general resources**
  - Extract specified profile - ADMN_XTR_RESOURCE (X'1F')
  - Extract next profile - ADMN_XTR_NEXT_RESOURCE (X'20')
- **Authorization required for problem state callers:**
  - At least READ access to the IRR.RADMIN.RLIST resource in the FACILITY class
  - Users are limited to seeing only the information that would be displayed by an RLIST command
    - For example , audit settings will be suppressed if caller does not have the AUDITOR attribute
- **Supervisor callers can request either, both, or no check**
  - Command authority enforced by default

## z/OS V1R11: R_admin Enhancements…

- **R_admin SETROPTS option extraction (ADMN_XTR_SETR (X'16')) may now be called from problem state**
- **Authorization required for problem state caller:**
  - At least READ access to IRR.RADMIN.SETROPTS.LIST in the FACILITY class
  - Authority as enforced by the SETROPTS command
    - For example, audit settings will be suppressed if caller does not have the AUDITOR attribute
- **No changes required to existing programs other than to remove MODESET into supervisor state**






## z/OS V1R11: LDAP Change Logging of General Resources

- **You can now tell RACF to create change log entries for changes to general resources by defining the profile NOTIFY.LDAP.*class-name* in the RACFEVNT class and activate the class**
- **Events which are logged:**
  - Resource additions made using the **RDEFINE** command
  - Resource modifications made using the **RALTER** command
  - Changes to the resource's access list using the **PERMIT** command
  - Resource deletions made using the **RDELETE** command
- **ICHEINTY/RACROUTE applications can create their own change log entries using R_proxyserv (IRRSPY00)**

## z/OS V1R11: REXX Interface to R_admin Extract Functions

- **IRRXUTIL allows you to extract information from the RACF database using the REXX programming language**
- **Data is returned as stem variables**

```
/* REXX */
myrc=IRRXUTIL("EXTRACT","USER","IBMUSER","RACF")
  if (word(myrc,1)=0) then do
    say "User ID  is "RACF.PROFILE
    say "Owner    is "RACF.BASE.OWNER.1
    say "UID      is "RACF.OMVS.UID.1
    say "Default grp is "RACF.BASE.DFLTGRP.1
    do i=1 to RACF.BASE.CGROUP.0
       say " Connect Group "i" "RACF.BASE.CGROUP.i
    end
  end
```

```
ex 'onghena.rrsf.clist(irrexx4)'
User ID  is IBMUSER
Owner    is IBMUSER
UID      is 0
Default grp is SYS1
 Connect Group 1 SYSCTLG
 Connect Group 2 SYS1
 Connect Group 3 VSAMDSET
READY
```






## z/OS V1R11: Automatic UID/GID Assignment

- **z/OS UNIX System Services tasks are associated with user and group identifiers (UIDs & GIDs)**
  - Can be assigned explicitly in RACF profiles (preferred)
    - AUTOUID/AUTOGID can be specified to generate a unique UID/GID
  - Can default from BPX.DEFAULT.USER profile
- **New option to assign permanent unique UIDs/GIDs is enabled by BPX.UNIQUE.USER profile in FACILITY class. Once enabled, RACF and UNIX System Services:**
  - Create a unique UID/GID and
  - Generates an OMVS segment for the user/group if none exists
    - APPLDATA specifies a default user profile from which the other segment information is copied.
  - Uses the existing BPX.NEXT.USER processing (from AUTOUID/AUTOGID)
- **Implemented in initUSP, getUMAP, & getGMAP, which are invoked by various UNIX system services**

## z/OS V1R11: Profile Name in Authorization Exits

- **The RACROUTE REQUEST=AUTH (ICHRCX02) and REQUEST=FASTAUTH(ICHRFX02,ICHRFX04) exits have always received a pointer to the profile which was used in the access control decision**
  - Profile is one which allowed or denied the request
  - Can differ from the resource name (if a generic profile was matched)
- **With z/OS V1R11, the exits receive the name of the profile as well**
  - For REQUEST=FASTAUTH, if the profile name is generic, then the internal format of the profile name is returned
  - RACROUTE REQUEST=AUTH, the profile name is always in external format
  - A new service is provided to map the internal format of the profile name to the external format






## z/OS V1R11: SMF Unload of WAS & TKLM SMF data

- **Support for z/OS SMF repository of audit data**
  - WebSphere Application Server (WAS) V7
  - Tivoli Key Lifecycle Manager (TKLM)
- **Both WAS and TKLM create SMF Type 83 records via the z/OS JAVA class interface to the R_auditx service**
  - Subtype 5: WebSphere
  - Subtype 6: TKLM
- **IRRADU00 unloads XML and traditional tabular-format output files of audit data**
- **LRECL of tabular IRRADU00 output is now 12288**
  - Will be adjusted automatically by IRRADU00

## z/OS V1R11: Digital Certificate Support

- **RACDCERT multi-byte character improvements**
  - Support (installation, retrieval and authentication) for certificates which contain characters which are outside the 1047 code page.
  - If a character does not map to code page 1047, the character will be represented by 6 characters in the format of U+nnnn, where nnnn is the Unicode code point of that character in hexadecimal format
  - When the certificate profile is created, the 6-character format will contribute to the profile name.
    - There is a risk of exceeding the profile name limit, which will prevent the creation of the certificate in RACF.
- **PKI Private Key recovery**
  - Prior to z/OS V1R11, PKI services did not generate private/public key pairs. In R11, key generation and key archival capabilities are being introduced. The certificate requestor will have the option to generate the public/private key pair themselves as in previous releases or have PKI Services generate the key pair.






## z/OS V1R11: Digital Certificate Support

- **PKI Web Pages**
  - PKI services now provides Java server pages (JSPs) and an XML template file to create and customize the PKI Services Web application as an alternative to the existing REXX CGI support.
- **PKI Support for SHA256 with RSA signature algorithm**
  - PKI Services will support the "SHA256 with RSA encryption" signature algorithm for signing certificates, certificate and authority revocation lists (CRL/ARL), and OCSP responses