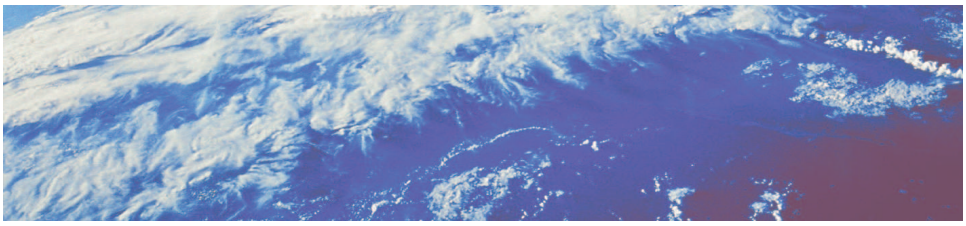
z/OS® RACF® Update 2012

Eric Rosenfeld CISSP®
z/OS Security Development
IBM Poughkeepsie
rosenfel@us.ibm.com

SHARE Atlanta
Session 10421
March 2012

Visit www.SHARE-SEC.com for more information on the SHARE Security & Compliance Project



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

What's new with z/OS V1.12 RACF?

- Generic Profile Loading
- SAFTRACE filtering by user ID or general resource class
- "Ghost" Generics
- Caller's Address in RACXTRT work area
- Support for ICSF

What's new with z/OS V1.12 Digital Certificate Support?

- Support for elliptic curve cryptography (ECC)
- Longer RSA keys
- DSA key types
- Extended certificate validity
- Certificate Management protocol support
- Configurable maintenance window

Agenda ...

What's new with z/OS V1.13 RACF & PKI

- RACF Remote Sharing (RRSF) over TCP/IP
- Identity Propagation extensions
- RACF Support for Elliptic Curve Cryptography (ECC)
- Support of DB2 for PKI Services Databases
- Larger Certificate Revocation Lists (CRLs)
- Enhanced support for Web Browsers

- **z/OS V1.13 Statement of Direction (RACF)**

- Click to add an outline

z/OS V1.12

z/OS V1.12: Generic Profile Load Performance

- **RACF caches up to 4 sets of generic profile names per address space to speed up authorization checks for resources which are covered by generic profiles**.
  - Known as **GATE**s (**G**eneric **A**nchor **T**able **E**ntries).
  - One per data set HLQ or general resource class that is neither SETROPTS RACLISTed, RACLISTed using RACROUTE REQUEST=LIST,GLOBAL=YES, or SETROPTS GENLISTed.
- **If an address space uses more than 4 sets of profiles RACF discards the least recently used list of generic profiles.**
- **If a deleted HLQ or class is referenced, the list is built again, which can result in thrashing.**
- **Prior to V1.12 what could you do?**
  - Split the RACF database
  - Physically rename data sets to reduce the number of generic profiles under a single HLQ.
  - Doing an analysis of the existing generic profiles to try to reduce their numbers
  - Implementing a RACF Naming Convention Table

z/OS V1.12: Generic Profile Load Performance …

- **With V1.12, you can configure the number of sets of profiles!**
  - Specified using the RACF SET command
  - Can be set system wide or by job name
  - Minimum: 4; Maximum: 99
  - A new TRACE operand has been added to the SET command to capture data about the caching of generic profiles to assist IBM support in diagnosing problems.

- **RACF has reorganized the way that GATEs are processed:**
  - Now in 64-bit storage (instead of ELSQA)
  - No longer searched sequentially, but it a hybrid manner:
    - Binary and sequential

z/OS V1.12: Generic Profile Load Performance …

- **The SET Command:**

```
SET …
     [ GENERICANCHOR(
     {SYSTEM | JOBNAME(jobname ...) }
     {COUNT( number ) | RESET })
     ]
```

- **SYSTEM** increases the number of generic profile caches system wide, for all jobs which do not have an overriding value

- **JOBNAME** increases the number of generic profile caches for all jobs which match the value specified. “*” may be used as a “don’t care” character at the end

- Additional tracing can be activated using this SET command:

```
SET … [ TRACE ( …{ GENERICANCHOR...} )
```

z/OS V1.12: Generic Profile Load Performance …

- **The SET LIST Command shows the status:**

```
RACFR12 IRRH005I (@) RACF SUBSYSTEM INFORMATION:
 TRACE OPTIONS - IMAGE
 - NOAPPC
 - SYSTEMSSL
 - RACROUTE
   2 5 9
 - NOCALLABLE
 - NOPDCALLABLE
 - NODATABASE
 - GENERICANCHOR (or NOGENERICANCHOR)
 ...
 PASSWORD SYNCHRONIZATION IS *NOT* ALLOWED
 AUTOMATIC DIRECTION OF APPLICATION UPDATES IS *NOT* ALLOWED
 GENERICANCHOR:
    SYSTEM: COUNT(nn)
    JOBNAME: job1 COUNT(nn)
    job2* COUNT(nn)
```


z/OS V1.12: SAFTRACE Filtering

- **The SAFTRACE facility, allows an in-depth analysis of the calls made from resource managers to RACF**
  - Can trace at the RACROUTE, callable service, or ICHEINTY level
  - Cannot instruct SAFTRACE to only trace for a specific class or specific user ID
  - Trace records are written to GTF and formatted with IPCS
  - Intended for use under the direction of RACF's service team
  - SET Syntax:

```
SET TRACE( …
          ASID(asid ... |*) | NOASID | ALLASIDS
            JOBNAME(jobname ... |*) | NOJOBNAME | ALLJOBNAMES
            RACROUTE(ALL | NONE | TYPE(type ...)) | NORACROUTE
            …
          )
```

z/OS V1.12: SAFTRACE Filtering by Class

- **With V1.12, you can control SAFTRACE records for RACROUTE and database (ICHEINTY) access by class:**

```
SET TRACE(CLASS(class … |*)      |
          IFCLASS(class …|*)     |
          NEVERCLASS(class …|*)  |
          NOCLASS)
```

- **CLASS is an inclusive setting**
  - Trace records which match CLASS are recorded.
  - Trace records which do not match CLASS() *may* be recorded if they match another setting, like ASID or JOBNAME.
- **IFCLASS is an exclusive setting**
  - Trace records which do not match IFCLASS are always discarded, even if they match other trace setting, like ASID or JOBNAME.
- **NEVERCLASS** discards all trace records whose class names match, regardless of other settings

z/OS V1.12: SAFTRACE Filtering by User ID

- **With V1.12, you can also control SAFTRACE records created for RACROUTE traces by user ID:**

```
SET TRACE(USERID(userid ... |*) |
          IFUSERID(userid …|*) |
          NEVERUSERID(userid …|*) |
          NOUSERID)
```

- **USERID is an inclusive setting:**
  - Trace records which match the user id are recorded
  - Trace records which do not match USERID() *may* be recorded if they match another setting, like CLASS or JOBNAME
- **IFUSERID is an exclusive setting:**
  - Trace records which do not match IFUSERID are always discarded. even if they match other trace setting, like CLASS or JOBNAME
- **NEVERUSERID** discards all trace records whose user id names match, regardless of other settings

z/OS V1.12: “Ghost” Generics

- **RACF requires that SETROPTS GENERIC is in effect for a class before generic profiles are defined in the class**
  - If not, the profile is created as a discrete profile which contains generic characters, such as “*”, “&”, or “%”
  - Profiles such as these are:
    - Not involved in access control decisions
    - Probably not what you intended
    - Displayed by SEARCH, RLIST, and LISTDSD without the “(G)” after the name
    - Require that you turn generics and GENCMD off for the class, delete the profile, SETROPTS GENERIC the class (which also turns GENCMD on), and redefine the profile
    - Annoyances to security administrators, systems programmers, and auditors

z/OS V1.12: “Ghost” Generics …

- **With V1.12, RACF now issues a warning message when creating a profile which contains generic characters (*,% or &) in a non-generic class**

```
ICH10321I The profile name profile_name contains generic
characters, but generics are not enabled for class
class_name.  A discrete profile has been created.
```

- **The message is suppressed for profiles in the RACFVARS class, in which discrete profiles with generic characters are intentionally created**

z/OS V1.12: “Ghost” Generics …

- **The RLIST command does not show existing ghost generic profiles, unless '*' is specified for the profile name**
- **The SEARCH command does display ghost generic profiles**
- **Both commands will now label ghost generic profiles as '(UNUSABLE)' in their output**

```
RLIST FACILITY T*

  CLASS      NAME
  -----      ----
  FACILITY   T* (UNUSABLE)    <--- ghost generic indicator
  ...

 CLASS NAME
 ----- ----
 FACILITY T* (G)  <-- Standard generic indicator
 ...
```


z/OS V1.12: “Ghost” Generics …

- **NOGENERIC keyword added to the RDELETE command to facilitate the deletion of ghost generic**

```
RDELETE FACILITY T* NOGENERIC
```

- **Specifies that you want RACF to delete the discrete profile**
  - If a generic profile with the same name exists, it will be unaffected.
- **SAF callable service R_admin also updated such that the Delete function supports a GENERIC=N flag**
- **RACF panels also support NOGENERIC processing** ...

z/OS V1.12: Caller's Address in EXTRACT Area

- **Applications for which RACF gets storage on a RACROUTE REQUEST=EXTRACT are required to free this storage when the application is finished with the data**
- **Ill-behaved applications which don't free this area can cause an out-of-storage condition**
  - It's difficult to identify the offending application/request as there is no information which ties the application to the storage
- **With V1.12, the callers ASID and return address are placed in the returned work area to assist in identifying the application which created the REQUEST=EXTRACT work area**
- **Mapped in EXTWKEA in IRRPRXTW**

z/OS V1.12: RACF Enhancement for ICSF

- **New keyword on ICSF segment on CSFKEYS, GCSFKEYS, XCSFKEY, and GXCSFKEY profiles allows the specification of controls on high performance secure keys**
  - **SYMCPACFWRAP([YES|NO])** Can this High Performance Secure key be exported?
- This new keyword has been added on the ICSF operand for the commands:
  - **RALTER**
  - **RDEFINE**
  - **RLIST**

z/OS V1.12: Digital Certificate Enhancements

- **Support for elliptic curve cryptography (ECC) when creating certificates and processing certificates created using ECC**
  - Complete SHA2 support (SHA224, SHA256, SHA384, SHA512)
  - Support for RACDCERT BIND and IMPORT on ECC and DSA keys
  - Support for ECC certificates and ECC keys from RACF key rings and PKCS#11 tokens using the R_datalib callable service

- **Support for creating RSA keys up to 4096 bits**

z/OS V1.12: Digital Certificate Enhancements ...

- **Support for long issuer distinguished names**
  - 246 character limitation for the issuer's distinguished name
  - Now 1024 character DNs for RACDCERT ADD and GENCERT, R_datalib, InitACEE, and PKI Services
  - Rolled back to z/OS V1.10 and V1.11
    - RACF APAR: OA30560
    - PKI APAR: OA30952

- **Extend certificate validity date beyond its current limit**
  (PKI Services: 2038, RACF:2041)
  - Extended to 12/31/9997
  - Supported by RACDCERT ADD, IMPORT, GENCERT, REKEY, LIST, and CHECKCERT and PKI Services
  - Rolled back to V1.10 add V1.11
    - RACF APAR: OA30560 (except RACDCERT GENCERT and REKEY)
    - PKI APAR: OA30952 (requires LE PTF UK47654 (v1.10), UK47655 (v1.11))

z/OS V1.12: PKI Services Enhancements

- **Support for certificate management protocol (CMP)**
  - CMP is the protocol that is used to manage X.509 certificates within a PKI-infrastructure. The support of these CMP in accordance with RFC 4210/4211 allows greater interoperability of z/OS PKI Services:
    - Certificate Request Message, type 2 (cr)
    - Certificate Response Message, type 3 (cp)
    - PKCS10 Certificate Request Message, type 4 (p10cr)
    - Revocation Request Message, type 11 (rr)
    - Revocation Response Message, type 12 (rp)
    - Error Message, type 23 (error)
- **Support for custom X.509 certificate extensions**
- **Support for the posting of certificates and certificate revocation lists (CRLs) to LDAP at any time**
- **Configurable maintenance task execution time**

- Click to add an outline

z/OS V1.13 RACF

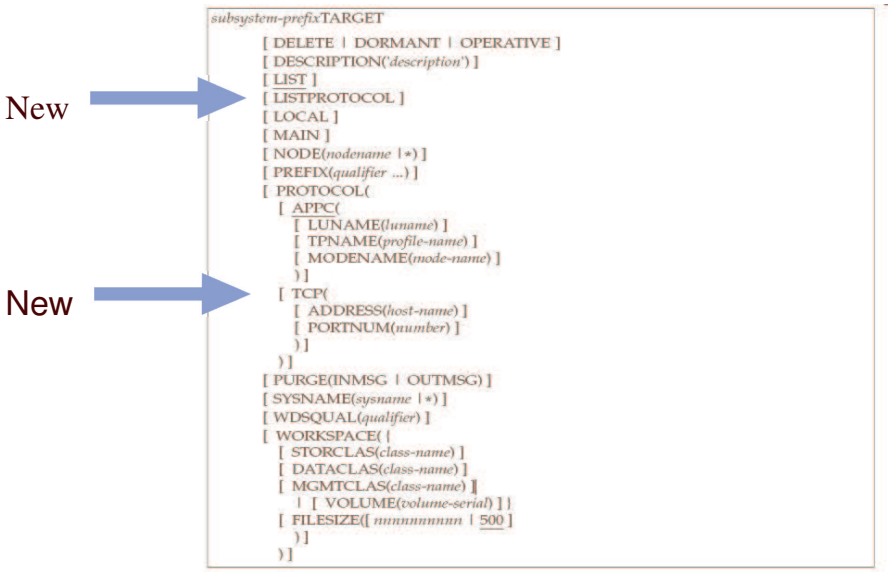
z/OS V1.13 RRSF over TCP/IP

- Problem:
  - Clients do not have the APPC and VTAM skills required to setup and maintain an RRSF network.
- Solution
  - RRSF will support TCP/IP (IPv4 only) as an alternate transport protocol.
- Benefit / Value
  - Clients already have the skills necessary to maintain a TCP/IP network.
    - Improve usability
    - Simplify network configuration
  - Stronger cipher algorithms are supported to protect the data while it crosses the network.

z/OS V1.13 RRSF...

- **RACF Remote Sharing Facility (RRSF)** over **TCP/IP …**
  - Now you can:
    - Manage your RRSF network using the same skills as the rest of your TCP/IP network.
    - Ensure that the same network security policy (IDS, IPS, etc.) is in place for your RRSF network as in place for the rest of your z/OS TCP/IP network.
    - Utilize the encryption and peer-node authentication of AT-TLS
    - Keep up with improvements in z/OS Communications Server Security.
    - Convert a node from using APPC to TCP/IP without stopping communication

z/OS V1.13 RRSF... Syntax of the TARGET command



TARGET command syntax: The LISTPROTOCOL and TCP keywords are new

z/OS V1.13 RRSF... Defining a TCP/IP node and activating it using TARGET (Simplified)

- The only difference from APPC is the PROTOCOL information:
  - Define the local node with a socket listener

```
TARGET NODE(LOCAL1) LOCAL PROTOCOL(TCP)
PREFIX(SYS1.RRSF) WORKSPACE(VOLUME(VOL001))OPERATIVE

IRRC054I (<) RACF REMOTE SHARING TCP LISTENER HAS BEEN SUCCESSFULLY
  ESTABLISHED.
```

  - Define the remote node and make it operative

```
TARGET NODE(REMOTE1) PROTOCOL(TCP(ADDRESS(remote.pok.ibm.com)))
PREFIX(SYS1.RRSF) WORKSPACE(VOLUME(VOL001)) OPERATIVE

IRRI027I (<) RACF COMMUNICATION WITH TCP NODE REMOTE1 HAS BEEN
  SUCCESSFULLY ESTABLISHED USING CIPHER ALGORITHM 35
  TLS_RSA_WITH_AES_256_CBC_SHA.
```

  - Harden your TARGET commands in the RACF parameter library

z/OS V1.13 RRSF...
TARGET LIST: summary version

- A new message line, prefixed with IRRM091I, indicates the status of each protocol listener defined to the local node.

```
- NODE1  <target list
- NODE1  IRRM009I (<) LOCAL RRSF NODE NODE1 IS IN THE OPERATIVE ACTIVE
-  STATE.
- IRRM091I (<)      - LOCAL NODE APPC LISTENER IS ACTIVE.
- IRRM091I (<)      - LOCAL NODE TCP LISTENER IS ACTIVE.
- IRRM009I (<) REMOTE RRSF NODE NODE2 IS IN THE OPERATIVE ACTIVE STATE.
```

- Status values are ACTIVE, INACTIVE, and INITIALIZING

z/OS V1.13 RRSF...
TARGET LISTPROTOCOL

- LISTPROTOCOL is a new keyword that displays the protocol in IRRM009I for remote nodes

```
- NODE1  <target listprotocol
- NODE1  IRRM009I (<) LOCAL RRSF NODE NODE1 IS IN THE OPERATIVE ACTIVE STATE.
- IRRM091I (<)      - LOCAL NODE APPC LISTENER IS ACTIVE.
- IRRM091I (<)      - LOCAL NODE TCP LISTENER IS ACTIVE.
- IRRM009I (<) REMOTE RRSF NODE NODE2 PROTOCOL TCP IS IN THE OPERATIVE ACTIVE STATE
- IRRM009I (<) REMOTE RRSF NODE NODE3 PROTOCOL TCP IS IN THE OPERATIVE ACTIVE STATE
- IRRM009I (<) REMOTE RRSF NODE NODE4 PROTOCOL APPC IS IN THE OPERATIVE ACTIVE STATE
- IRRM009I (<) REMOTE RRSF NODE NODE5 PROTOCOL TCP IS IN THE OPERATIVE ACTIVE STATE
- IRRM009I (<) REMOTE RRSF NODE NODE6 PROTOCOL APPC IS IN THE OPERATIVE ACTIVE STATE
```

- Comes in handy when displaying a mixed-protocol network

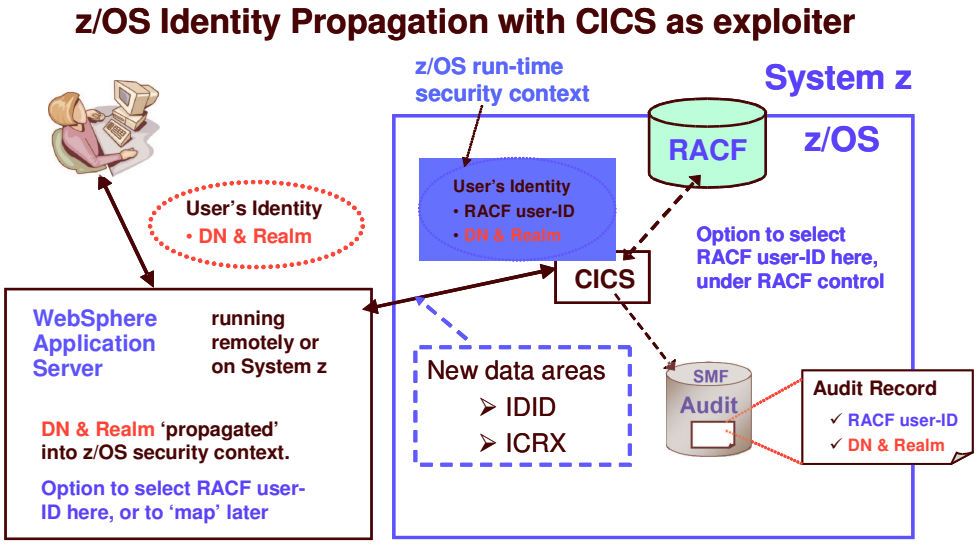
z/OS V1.13: Identity Propagation Overview

- z/OS Identity Propagation, introduced with z/OS V1R11, is extended by implementing enhancements that are required by exploiters of this function.

- With this support you can achieve end-to-end security identity consistency and auditing for key system environments such as:
  - **CICS**
  - **DB2**
  - **WAS**
  - **DataPower**

- This provides **consistent end-to-end auditing** of z/OS transactions that originate from the Internet by maintaining the user's distributed identity information, without impacting the performance characteristics of transaction providers.

z/OS V1.13: Identity Propagation... Usage & Invocation



z/OS V1.13: Identity Propagation... Overview

- The enhancements to z/OS Identity Propagation function introduced in z/OS V1R11 are the following:
  - **R_usermap callable service provides a query service**
    - Will take a Distinguished Name (DN) and a Registry/Realm Name and return the matching RACF user ID.
  - **RACMAP command provides a query function**
    - Use the UserDIDfilter Name and Registry Name to return the matching RACF user ID.
  - RACLIST function, for the IDIDMAP class, enhanced to reduce I/O when the same User's Distinguished Name is defined for more than one registry
  - **R_cacheserv callable service** updated
    - **Provides a service to validate an ICRX containing an IDID with section 1 completed.**
    - **Allows reusable ICRX objects.**
  - **Normalizes the Distributed Identity Filter Name if it is in X.500 format.**

z/OS V1.13: Identity Propagation …

- **Identity Propagation Extensions**
  - The RACMAP command now provides a query function which uses the UserDIDfilter Name and Registry Name and returns the matching RACF user ID.

Syntax:

```
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'))]
        | QUERY
            USERDIDFILTER(NAME('distributed-identity-username-filter'))
            REGISTRY(NAME('distributed-identity-registryname'))
```

z/OS V1.13: Identity Propagation... Usage & Invocation

- Enhancements to the following existing interfaces:
  - **R_usermap** has been enhanced with
    - A new **Function Code** (**8**) and
    - Two new parameters:
      - **Distinguished_Name**
      - **Registry Name**

```
CALL IRRSIM00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Function_code,
               Option_word,
               RACF_userid,
               Certificate,
               Application_userid,
               Distinguished_Name,
               Registry_Name
              )
```


z/OS V1.13: Identity Propagation...

- Distinguished Name matching updated
  - Removes leading/trailing nulls (in addition to blanks) from the Distinguished Name (DN) and Registry Name.
  - Normalizes the Distinguished Name, if it is in X.500 format, prior to
    - Storing the filter name in the RACF DB.
    - Matching the Distinguished Name to the extracted filter name.
  - Normalization inspired by RFC 4514 (which supersedes 2253), but not fully implemented
    - Differs from rules in the field, so a utility is provided in SAMPLIB to see if you have affected filters.
  - Rules of normalization documented under the RACMAP command in the Command Language Reference

z/OS V1.13: Identity Propagation …

- Rolled back to R11 via:
  - SAF APAR **OA34259** (PTF's **UA59871** for R12 and **UA59870** for R11)
  - RACF APAR **OA34258** (PTF's **UA59873** for R12 and **UA59872** for R11)
- Available now, based on RACF function in z/OS V1.11
  - **CICS Transaction Server 4.1**
  - **DataPower 3.8.0**
  - **DB2 V10**
  - **WAS V7 via fixpack**
  - **WAS for z/OS Optimized Local Adapters (WOLA) V8**

z/OS V1.13: RACDCERT

- **RACF support for hardware-generated Elliptic Curve Cryptography (ECC) secure keys**,
  - Provides the ability to issue and use certificates with hardware-protected ECC keys**.**
  - Exploits Crypto Express 3 Cryptographic Coprocessor (CEX3C)
  - New keywords on the RACDCERT command to allow users to specify that an ECC key be stored in the ICSF PKA key data set (PKDS).
  - Corresponding hardware ECC key support for PKI Services
- **Dependencies**
  - ICSF web deliverable #10
  - Crypto Express3 Coprocessor (CEX3C) card on IBM zEnterprise server.

z/OS V1.13: RACDCERT ...

- RACDCERT command and panels
  - New sub keyword PKDS is added to indicate the key is a hardware key. For examples:
    - Generate a certificate with NIST ECC key stored in PKDS with system generated key label
      - RACDCERT GENCERT SUB(CN('Company A')) WITHLABEL('New NISTECC cert') **NISTECC(PKDS)**
    - Generate a certificate with Brainpool ECC key stored in PKDS with key label BPECCFORA
      - RACDCERT GENCERT SUB(CN('Company A')) WITHLABEL('New BPECC cert') **BPECC(PKDS(BPECCFORA))**
  - Panels and corresponding help panels for GENCERT and REKEY will be updated to handle the new types.

- R_datalib callable service
  - New private key types X'00000009' (ECC key token), will be handled by functions DataGetFirst and DataGetNext

- PKI Services IKYSETUP (A REXX script to set up authorization for PKI)
  - Update the key_type value to 6 for hardware NISTECC, 7 for hardware BPECC

z/OS V1.13: RACDCERT ...

- Naming of key types has been restructured
  - For example, key types Non-ICSF, ICSF, PCICC are all RSA keys
  - Adding more key types worsens the situation
  - There are 4 public/private key types supported in RACF
    - RSA, DSA, NIST ECC, Brainpool ECC
    - Except for DSA, you can choose to generate/store the key in ICSF PKDS protected by the Master Key (hardware-protected)
  - Both key type and the place where it is stored is used to name the key
    - For example
      - NISTECC means key is stored in software
      - NISTECC(PKDS) means key is stored in hardware
- Advantage: Key types are more intuitive and more consistent for input and output

z/OS V1.13: RACDCERT ...

INPUT KEY TYPE IN RACDCERT GENCERT	KEY TYPE DISPLAYED IN RACDCERT LIST
NISTECC	Key Type: NISTECC (no PKDS label entry)
NISTECC(PKDS)	Key Type: NISTECC PKDS Label: <system generated label>
NISTECC(PKDS(<specified label>))	Key Type: NISTECC PKDS Label: <specified label>
BPECC	Key Type: BPECC (no PKDS label entry)
BPECC(PKDS)	Key Type: BPECC PKDS Label: <system generated label>
BPECC(PKDS(<specified label>))	Key Type: BPECC PKDS Label: <specified label>
RSA = no key type specified	Key Type: RSA (no PKDS label entry)
RSA(PKDS) = PCICC	Key Type: RSA PKDS Label: <system generated label>
RSA(PKDS(<specified label>))	Key Type: RSA PKDS Label: <specified label>

- Click to add an outline

z/OS V1.13 PKI Services

z/OS V1.13 PKI Services

- Support of DB2 for PKI Services Databases
  - Both PKI Services databases can be stored in DB2
    - Object store – holds records to track active certificate requests and posting objects for certificates and CRLs
    - Issued Certificate List – permanent record for each certificate issued
  - Enables enterprise-class scale and resilient certificate management
    - Issued certificate list (ICL) can contain millions of certificates
  - Utilities provided to migrate existing PKI data from VSAM to DB2
  - DB2 9 for z/OS or later is required

z/OS V1.13 PKI Services...

- Larger Certificate Revocation Lists (CRLs)
  - When LDAP posting is enabled, 32K limit is removed by storing the CRLs in the HFS or zFS instead of in VSAM
- Enhanced support for Web Browsers
  - Previously, PKI Services only supported IE to use a smart card for the Windows Logon certificate generation
  - Now Mozilla-based browsers on both the Windows and Linux platforms can use a smart card to generate certificates

- Click to add an outline

z/OS V1.13 Statement of Direction (RACF)

z/OS V1.13 Statement of Direction (RACF) …

- **Background: Assigning UID and GIDs**
  - **RACF 2.1 (1994):** Introduced OMVS segments for USERs and GROUPs.
    - Users with an OMVS segment could now use “Open MVS” (now z/OS UNIX System Services)
  - **OS/390 R2.4 (1997**): Introduced BPX.DEFAULT.USER FACILITY class profile
    - Allows assigning UIDs and GIDs to users and groups who do not have OMVS segments;

      One UID and one GID shared by all default users

z/OS V1.13 Statement of Direction (RACF) …

- **Background: Assigning UID and GIDs…**
  - **z/OS V1.4 (2002):** Introduced AUTOUID/AUTOGID keyword on ADDUSER, ALTUSER, ADDGROUP, ALTGROUP
    - RACF could now find the next available UID or GID using the BPX.NEXT.USER profile in the FACILITY class
    - Required enabling RACF Alternate Index Mapping ("AIM") to stage 2
      - Limitation of 129 eight-character users sharing one UID
      - Required running migration utility ("IRRIRA00")
  - **z/OS V1.11 (2009):** Automatic generation of OMVS segment for USERs and groups
    - Built upon AUTOUID/AUTOGID
    - Requires AIM stage 3
    - Uses the BPX.UNIQUE.PROFILE in the FACILITY class

Statement of Direction

- From *Preview: z/OS Version 1 Release 13 and z/OS Management Facility Version 1 Release 13 are planned to offer new availability, batch programming, and usability functions*
  - IBM United States Software Announcement 211-007
  - February 15, 2011
  - z/OS V1.13 is planned to be the last release to support BPX.DEFAULT.USER. IBM recommends that you either use the BPX.UNIQUE.USER support that was introduced in z/OS V1.11, or assign unique UIDs to users who need them and assign GIDs for their groups.

Publications

- *z/OS Security Server RACF Security Administrator's Guide*
  - Section "*Automatically assigning unique IDs through UNIX services*"

z/OS V1.13 Statement of Direction (RACF) …

- **What this means to you:**

1. If you are using BPX.UNIQUE.USER then:
   - You are not using BPX.DEFAULT.USER (even if it is defined)
   - This SoD has no impact to you.

1. If you are already assigning UIDs and GIDs to all users using z/OS UNIX System Services by assigning OMVS segments to all necessary users and groups, then:
   - You must continue to assign all new users and groups OMVS segments

1. If you are already assigning UIDs and GIDS to all users using z/OS UNIX System Services by defining OMVS segments using AUTOUID/AUTOGID (which uses BPX.NEXT.USER) then:
   - You are already using AIM at a minimum of stage 2
   - You must continue to assign all new users and groups OMVS segments

1. If you are using only BPX.DEFAULT.USER
   - You must either move to the automatic generation of OMVS user and group segments or assign OMVS user and group segments to all necessary users and groups

Helpful Product Publications

- **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**
- **SA22-7681 - z/OS Security Server RACF System Programmer's Guide**
- **SA22-7692 - z/OS Security Server RACROUTE Macro Reference**
- **GA22-7689 - z/OS Security Server RACF Diagnosis Guide**
- **SA22-7693 - z/OS Cryptographic Services PKI Services Guide and Reference**
- **SC24-5901 - z/OS Cryptographic Services System Secure Sockets Layer Programming**
- **SA23-2231 - z/OS ICSF Writing PKCS #11 Applications**
- **SA22-7807 - z/OS UNIX System Services: Messages and Codes**
- **SA22-7803 - z/OS UNIX System Services Programming: Assembler Callable Services Reference**
- **SC31-8775 - z/OS Communication Server: IP Configuration Guide**
- **GC31-8782 - z/OS Communication Server: IP Diagnosis Guide**
- **SC31-8781 - z/OS Communication Server: IP System Administrator's Commands**

Helpful References

- **IBM Redbooks**
  - **z/OS V1 R8 RACF Implementation (SG24-7248)**
- **RFCs**
  - **RFC2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile**
  - **RFC5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile**
  - **RFC4210 - Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)**
  - **RFC4211 - Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)**

Questions, comments ?

- Backup Slides

RACF Access Control Module for DB2 for z/OS Version 10

DB2 V10: New DB2 System Authorities

- **DB2 for z/OS Version 10 introduces new system authorities that allow for a finer granularity of control:**
  - **SECADM:** Manage all of the security-related objects in DB2 and control access to all DB2 resources in native DB2 security
  - **System DBADM:** Manage most objects in a DB2 subsystem, without having the ability to access data or control access to data
  - **DATAACCESS:** Access data in all user tables, materialized query tables, and views and execute plans, packages functions and procedures in a DB2 subsystem.
  - **ACCESSCTRL:** Grant all authorities and privileges except, DBADM, DATAACCESS, ACCESSCTRL and privileges on security-related objects.
  - **SQLADM:** Monitor and tune DB2 without have any other privilege

DB2 V10: New DB2 System Authorities …

- **If you are using the RACF Access Control Module for DB2 (DSNXRXAC) you can grant these authorities by giving a user READ authority to these resource names in the indicated class:**

DB2 Authority	RACF General Resource Class	Resource Name
ACCESSCTRL	DSNADM	*db2-subsystem*.ACCESSCTRL
DATAACCESS	DSNADM	*db2-subsystem*.DATAACCESS
EXPLAIN	DSNADM	*db2-subsystem*.EXPLAIN
SECADM	DSNADM	*db2-subsystem*.SECADM
SQLADM	MDSNSM	*db2-subsystem*.SQLADM
System DBADM	DSNADM	*db2-subsystem*.SYSDBADM

DB2 V10: Other New Security Functions

- **Separation of Duties**
  - You can configure DB2 to prevent users with SYSADM authority from altering authorizations, thus restricting security-related work to SECADM users.
  - This is done by setting the “SEPARATE SECURITY” ZPARM to ‘YES’
  - When SEPARATE_SECURITY is set to ‘YES’, then the SYSADM and SYSCTRL authorities cannot be used to affect the security characteristics of the system. Specifically:
    - The SYSADM authority does not allow the management of security objects, such as roles and trusted contexts.
    - The SYSCTRL authority does not allow the management of roles.
    - The SYSADM and SYSCTRL authorities cannot perform grants and cannot revoke privileges granted by others.
- **Row and Column Access**
  - DB2 allows you to restrict access to the contents of a table by row by and column
- **Significant logging enhancements**