RACF® Update

Julie Bergh

Agenda

- **Common Criteria Evaluation Update**
- **z/OS V2.2 RACF Enhancements**
  - Read-Only Auditor
  - RRSF Enhancements
  - Enhancements for z/OS UNIX
  - Password Enhancements specific to z/OS V2.2
  - Require READ authority only for IRRDBU00 input data set if PARM=NOLOCKINPUT is specified
  - Certificate Enhancements
  - RACF Support for IBM Multi-Factor Authentication
- **Pre-V2.2 Enhancements**
  - New Health Checks
- **Other z/OS V2.2 Enhancements**
  - SMF Record Signing
  - DFSMSdfp BYPASS_AUTH
  - DFSMSdfp Erase-on-Scratch Enhancements

Common Criteria Update

Common Criteria Update

- **Recent Common Criteria Evaluations of Interest:**
  - z/OS V2.1/RACF, EAL5+, 14 April, 2015
  - z/OS V2.1, EAL4+, 2 September 2014
  - z/OS V1.13, EAL4+, 12 September, 2012
  - z/OS V1.13/RACF, EAL5+, 27 February, 2013
  - z/VM Version 6 Release 3, EAL4+, 30 March, 2015
  - z/VM Version 6 Release 1, EAL4+, 20 February, 2013
  - PR/SM for IBM zEnterprise EC13 GA1, EAL 5+, 15 October, 2015
  - PR/SM for IBM zEnterprise EC12 GA2/BC12 GA1 EAL5+, 19 February, 2014
  - PR/SM for IBM zEnterprise EC12 GA1 EAL5+, 19 February, 2013

- **z/OS V2.2 and RACF are in evaluation**

- **http://www.ibm.com/security/standards/security_evaluations.html has the details**

Read-Only Auditor

Read-Only Auditor

- **The read-only auditor (ROAUDIT) user attribute grants the user the ability to perform all of the activities of a user with the AUDITOR attribute except the ability to:**
  - Change profile content (such as AUDIT/GLOBALAUDIT settings)
  - Change system logging options
- **ROAUDIT allows a user to list all information about any RACF profile without needing to grant that user additional authority to those profiles**
- **Suitable for use by an external auditor who may need to verify the current security state of a system**
- **Assigned to users with the ADDUSER and ALTUSER commands**

Read-Only Auditor…

- **Syntax:**
  - `ADDUSER <user_ID> ROAUDIT`
  - `ALTUSER <user_ID> ROAUDIT`
  - `ALTUSER <user_ID> NOROAUDIT`
- **The RACF "list" commands honor ROAUDIT**
  - `LISTDSD, LISTGRP, LISTUSER, RLIST, SETROPTS LIST, SEARCH`
- **z/OS UNIX ck_access honors ROAUDIT**
- **The RACF utilities that honor ROAUDIT: DSMON, IRRUT100, and IRRXUTIL**
- **If ROAUDIT and AUDITOR are set, AUDITOR attribute takes precedence**

RRSF Enhancements

RRSF: Dynamic Main Switching

- **RACF’s Remote Sharing Facility (RRSF) allows the definition of multisystem nodes (MSNs) which are collections of systems which use shared DASD to share a RACF database.**
- **Switching the MAIN system in a multisystem node is a challenging “11”-step manual processes that is not feasible for short-term changes**
- **RRSF Dynamic Main Switching allows you to replace this onerous process with a single command**
  - Allows you to avoid even minor outage windows
  - Allows you to move RRSF workload off of a busy system
  - New programming interfaces introduce possibility of automating the switch entirely

RRSF: Dynamic Main Switching…

The “Dreaded 11-Step Process”:

1) Drop TSO/E and JES on the original local main system.
2) On the original local main system, issue the RACF STOP command to stop the RACF subsystem.
3) Make connections dormant:

    1) On the local system that is to be the new main, issue a TARGET DORMANT command for its local connection. Also issue TARGET DORMANT commands to make all connections with remote nodes dormant.

    2) On each remote node, issue TARGET DORMANT commands for the original and new main systems. Do not perform step 7 until the INMSG files for the original and new main systems on each remote node have drained.

Issue TARGET LIST commands to verify that the INMSG data sets on the local node have been drained before you go on to the next step.

1) If the workspace data sets for the original main system and the new main system are not on shared DASD with a shared catalog, copy the workspace data sets for the original main system to DASD accessible to the new main system, using the same workspace data set names.

2) On the new main system, issue a TARGET MAIN command to make it the main system. If you have not specified the prefixes for the workspace data sets and the LU names for the member systems consistently in the TARGET commands that defined the local multisystem node, this step will fail.

3) Issue the same TARGET MAIN command that you issued in step 5 on each nonmain system on the local multisystem node. Issue this command on the original main system only if it is to remain in the multisystem node.

4) Issue TARGET LIST commands to verify that the INMSG data sets on the remote nodes have been drained before you perform this step. On each remote system (that is, all remote systems of all remote nodes), issue the same TARGET MAIN command that you issued in step 5.

5) On the new main system, issue TARGET OPERATIVE commands to make the connection with itself and all connections with remote nodes operative.

6) On each remote system (that is, all remote systems of all remote nodes), issue TARGET OPERATIVE commands for the original main (if it is to remain in the multisystem node) and new main systems.

7) Update the TARGET commands in the RACF parameter libraries for all systems on all nodes in the RRSF network to reflect the new main system. If you fail to update the RACF parameter library for a system, the next time that system has its RACF subsystem restarted or is IPLed, the original TARGET commands will be issued, and requests and returned output will accumulate in the wrong OUTMSG workspace data set. However, RACF will issue appropriate error messages and prevent communications.

8) If the original main system is still part of the multisystem node, (and assuming that you have updated its RACF parameter library as discussed in step 10) restart the RACF subsystem, TSO/E and JES on the original main system

RRSF: Dynamic Main Switching…

- **When the Multisystem Node is in a sysplex, from any system in the multisystem node, issue:**

  ```
  TARGET NODE(msn-name) SYSNAME(new-main) PLEXNEWMAIN
  ```

- **RACF confirms the change with the message:**

  ```
  IRRM110I SYSTEM new-main HAS REPLACED SYSTEM old-
  main AS THE MAIN SYSTEM IN LOCAL NODE msn-name
  ```

- **Optionally, update the RACF parameter library to “harden” the change**

RRSF: Dynamic Main Switching...

- **When the Multisystem Node is not in a sysplex, from the current MAIN system in the multisystem node, issue:**

```
TARGET NODE(msn-name) SYSNAME(new-main) NEWMAIN
```

- **RACF confirms the change with the messages:**

```
IRRM098I DRAINING SYSTEM OF INBOUND WORK. DO NOT
INITIATE THE MAIN SWITCH ON THE NEW MAIN SYSTEM
UNTIL MESSAGE IRRM099I IS ISSUED

IRRM099I ALL INBOUND WORK HAS COMPLETED. IT IS NOW
SAFE TO INITIATE THE MAIN SWITCH ON THE NEW MAIN
```

RRSF: Dynamic Main Switching…

- **From the new MAIN system, issue:**

```
TARGET NODE(msn-name) SYSNAME(new-main) NEWMAIN
```

- **RACF confirms the change with the message:**

```
IRRM102I SYSTEM new-main IS NOW THE MAIN SYSTEM IN
LOCAL NODE msn-name.
```

- **From the remaining peer systems, issue:**

```
TARGET NODE(msn-name) SYSNAME(new-main) NEWMAIN
```

- **Optionally, update the RACF parameter library to “harden” the change**

- **Note that only z/OS V2.2 systems support dynamic main switching**

RRSF: Unidirectional Connections

- **When two systems are connected using RRSF, it is impossible to prevent a privileged user on one system from escalating his privilege on the other system.**
  - This issue is exacerbated if one system is a “test” system
- **With Unidirectional RRSF connections, one RRSF node can define another RRSF node such that inbound requests from that node are denied**
  - This can help protect against accidental or malicious damage to your production system
  - You can demonstrate to an auditor your compliance with your security policy, regardless of the configuration established on the remote node

RRSF: Unidirectional Connections…

- **The DENYINBOUND keyword on the TARGET command is used to reject commands from the specified node:**

```
TARGET NODE(thatnode) DENYINBOUND
```

- **When the remote node is a multisystem node:**

```
TARGET NODE(thatnode) SYSNAME(*) DENYINBOUND
```

- **SYSNAME(*) is not required as RACF will ensure that the setting is consistent across all systems when a single SYSNAME is changed.**

- **To change your mind, use ALLOWINBOUND.**
  - This is the default, so you don't need to code it in the parameter library

- **DENYINBOUND is ignored if specified for the LOCAL node**

Enhancements for z/OS UNIX

z/OS UNIX: Search Authority

- **When opening a file in the z/OS UNIX directory, the user must have READ and SEARCH authority on all directories in the path to the file**
  - Even if the user has an administrative authority such as SUPERUSER.FILESYS.CHANGEPERMS
  - Many installations have just granted those users a higher-than-desired authority such as AUDITOR or SUPERUSER.FILESYS
- **READ authority to the resource SUPERUSER.FILESYS.DIRSRCH in the UNIXPRIV class grants the user read and search permissions on z/OS UNIX directories**
  - Does not grant read, write, or execute permission to ordinary z/OS UNIX files
  - Does not grant write permission to z/OS UNIX directories
  - Generic profiles are supported

UNIX Search Authority

- **UNIX Security Administration:**
  - z/OS UNIX defines a set of **UNIXPRIV** class profiles to manage various UNIX privileges:

    SUPERUSER.FILESYS.CHANGEPERMS – Change file permissions
    SUPERUSER.FILESYS.CHOWN - Change file owners
  - These privileges lack the ability to read or search directories.
  - In order to search directories, the administrator must be granted one of:
    - Search authority to containing directories
    - RACF Auditor
    - **BPX.SUPERUSER** in **FACILITY** Class / UID 0
- **New V2R2 UNIX Search Authority:**
  - New function introduced in V2R2 will allow for directory read / search authorization to be granted via a new RACF profile:

    SUPERUSER.FILESYS.DIRSRCH - Allows a user to read and search all directories, without the authority to open other files.

UNIX Search Authority

- Allowing z/OS UNIX users to search directories:
  - To allow z/OS UNIX users to read and search all file system directories, regardless of file permission bits or access lists, create a profile in the **UNIXPRIV** class protecting a resource called **SUPERUSER.FILESYS.DIRSRCH**. Then permit users and groups with at least READ access performing the following steps.

1. Define a profile in the **UNIXPRIV** class.

```
RDEFINE UNIXPRIV SUPERUSER.FILESYS.DIRSRCH UACC(NONE)
```

2. Add the user or group to the access list with at least READ access.

```
PERMIT SUPERUSER.FILESYS.DIRSRCH CLASS(UNIXPRIV) ID(USER01 GRPX)
                ACCESS(READ)
```

3. If the **UNIXPRIV** class is not already active, activate and RACLIST it.

```
SETROPTS CLASSACT(UNIXPRIV) RACLIST(UNIXPRIV)
```

4. If the **UNIXPRIV** class is already active and RACLISTed, refresh it.

```
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

You have now given directory search permission to the specified users and groups.

z/OS UNIX: FSEXEC Control

- **Using profiles in the new FSEXEC class, installations can prevent the execution of files within the file system**
  - Profile name must match the FILESYSTEM name specified on the MOUNT statement
  - Profile name is case sensitive
  - Generic profiles are supported
  - Useful for directories such as /tmp where any user can write files
- **Example:**

```
RDEFINE FSEXEC /tmp UACC(NONE)
               or
RDEFINE FSEXEC OMVS.ZFS.ADMIN.** UACC(NONE)
PERMIT OMVS.ZFS.ADMIN.** CLASS(FSEXEC) ID(FRED) ACC(UPDATE)
SETR RACLIST(FSEXEC) REFRESH
```

z/OS UNIX: FSEXEC Control…

- **SUPERUSER or AUDITOR does not override FSEXEC denial of access**
- **FSEXEC is supported for zFS and tFS file systems**
- **FSEXEC does not apply to file systems mounted with the ‘-s nosecurity option’**
- **On denial, ICH408I message text includes ‘ACCESS ALLOWED (FSEXEC ---)’**

Password Enhancements for z/OS V2.2

Password Enhancements for z/OS V2.2

- **You never need an ICHDEX01 exit unless you are implementing your own password algorithm**
- **RACF_ENCRYPTION_ALGORITHM Health Check raises an exception if KDFAES is not active**
- **ADDUSER will not assign a default password**
  - `ADDUSER STU TSO(...) OMVS(...) NAME('DISCO STU')`
    - … now shows `ICH01024I User STU is defined as PROTECTED.`
  - ALTUSER and PASSWORD cannot be used to reset a password to the user's default group. It can, of course, be explicitly assigned...if your rules allow it!
- **RACLINK DEFINE(*node.user/pwd*) supports password phrases**
- **The RACF ISPF panels support the new OA43999 functions**

RACF Password Enhancements

RACF Password Enhancements

- **RACF Password Special Characters**
  - Additional characters now supported for passwords
- **Password Phrase only users**
  - Ability to have a password phrase without a password
- **Expire a password without setting a new password**
  - Ability to expire a password without having to set a new password
- **New Password Encryption option**
  - Optionally encrypt / hash the password and password phrase using a new more modern algorithm

RACF Password Special Characters

RACF Password Special character support

- **Support 14 additional characters**
  - Currently, there are 65 possible password characters if mixed-case is in effect
    - 65**8 =   318,644,812,890,625 possible 8-char passwords
  - With the additional 14 characters
    - 79**8 =1,517,108,809,906,561

RACF Password Special character support

Symbol	Hexadecimal value*
.	4B
<	4C
+	4E
|	4F
&	50
!	5A
*	5C
-	60
%	6C
_	6D
>	6E
?	6F
:	7A
=	7E

Usage & Invocation

- **Enable special characters with:**

  SETROPTS PASSWORD(SPECIALCHARS)

- **Confirm this with SETROPTS LIST:**

```
PASSWORD PROCESSING OPTIONS:
  THE ACTIVE PASSWORD ENCRYPTION ALGORITHM IS KDFAES
  PASSWORD CHANGE INTERVAL IS  60 DAYS.
  PASSWORD MINIMUM CHANGE INTERVAL IS   3 DAYS.
  MIXED CASE PASSWORD SUPPORT IS IN EFFECT
  SPECIAL CHARACTERS ARE ALLOWED.
  10 GENERATIONS OF PREVIOUS PASSWORDS BEING MAINTAINED.
  AFTER   5 CONSECUTIVE UNSUCCESSFUL PASSWORD ATTEMPTS,
      A USERID WILL BE REVOKED.
  PASSWORD EXPIRATION WARNING LEVEL IS  15 DAYS.
  INSTALLATION PASSWORD SYNTAX RULES:
    RULE 1  LENGTH(8)     xxxxxxxx
   LEGEND:
    A-ALPHA C-CONSONANT L-ALPHANUM N-NUMERIC V-VOWEL W-NOVOWEL *-
ANYTHING
    c-MIXED CONSONANT m-MIXED NUMERIC v-MIXED VOWEL $-NATIONAL s-
SPECIAL
    x-MIXED ALL
```

Usage & Invocation

- **Disable special characters with:**
  - SETROPTS PASSWORD(NOSPECIALCHARS)

  - Confirm this with SETROPTS LIST

```
PASSWORD PROCESSING OPTIONS:
    ...
SPECIAL CHARACTERS ARE NOT ALLOWED.
...
```

- **If the user has special characters in his password when the function is disabled:**
  - The user will always be able to logon with it and change their password (at logon or using the PASSWORD command) on that system

Password Phrase
Only Users

Usage & Invocation

- Uses existing **NOPASSWORD** keyword of **ADDUSER/ALTUSER**
- It is simply allowed now in cases where it previously wasn't
  - While specifying **PHRASE()**
  - When a phrase exists in the user profile
- Displayed by LISTUSER using existing attributes

```
USER=PHRONLY  NAME=UNKNOWN  OWNER=IBMUSER
  CREATED=14.206

DEFAULT-GROUP=SYS1     PASSDATE=N/A    PASS-
  INTERVAL= 30 PHRASEDATE=00.000

ATTRIBUTES=NOPASSWORD PASSPHRASE
```

- FLAG7 field in USER profile can now have both bit 0 (no password) and bit 2 (has password phrase) on

Expire Password
without setting a password

Expire a password and phrase without changing it

- Uses existing **EXPIRED** keyword of **ALTUSER**
- When specified without **PASSWORD** and **PHRASE**, sets the changed-dates to 0, thus expiring the password
  - Previously, **EXPIRED** was ignored in this situation
- **Before:**

```
USER=GRONK  NAME=UNKNOWN  OWNER=IBMUSER   CREATED=14.206

DEFAULT-GROUP=SYS1     PASSDATE=14.206 PASS-INTERVAL= 30
  PHRASEDATE=14.206

ATTRIBUTES=PASSPHRASE

ALTUSER GRONK EXPIRED
```

- **After:**

```
USER=GRONK  NAME=UNKNOWN  OWNER=IBMUSER   CREATED=14.206

 DDEFAULT-GROUP=SYS1     PASSDATE=00.000 PASS-INTERVAL= 30
  PHRASEDATE=00.000

 ATTRIBUTES=PASSPHRASE
```

New Password Encryption Algorithm

New Password Encryption Algorithm Option

- **RACF and Passwords:**
  - Passwords and Password phrases are encrypted / hashed before being stored in the RACF database.
- **DES Algorithm:**
  - Password / Phrase is run though a Key Derivation Function to generate a DES key. The DES key is used to encrypt the USERID.
  - Encrypted password hash can not be decrypted.
  - Passwords are validated by performing the password encryption algorithm on the user provided password and comparing to the encrypted password in the RACF database.
- **Offline database attack:**
  - Any password database should be protected to prevent any read attempt.
  - An attacker with access to an offline password database may attempt to recover passwords by systematically encrypting a large number of passwords and comparing to the encrypted values in the database.

New Password Encryption Algorithm

- **Start with:**
  - DES hash for passwords – Maintains upward compatibility
  - Clear-text password phrase
- **Append random text (salt)**
- **Iteratively hash (SHA256) this text a (large) number of times to derive an AES key**
  - This step is intentionally slowing down the encryption process!
- **Encrypt the RACF user ID with the AES key**
- **Results:**
  - 16-byte hash which must be stored with the salt and other information. This no longer fits in PASSWORD field. Extension fields are defined to contain the extra information.

Enabling the New Algorithm

- **Enable the new algorithm:**

```
SETROPTS PASSWORD(ALGORITHM(KDFAES))
```

  - No ICHDEX01 exit required to enable it!

- **Confirm this with SETROPTS LIST:**

```
PASSWORD PROCESSING OPTIONS:
  THE ACTIVE PASSWORD ENCRYPTION ALGORITHM IS KDFAES
  PASSWORD CHANGE INTERVAL IS  60 DAYS.
  PASSWORD MINIMUM CHANGE INTERVAL IS   3 DAYS.
  MIXED CASE PASSWORD SUPPORT IS IN EFFECT
  SPECIAL CHARACTERS ARE ALLOWED.
  10 GENERATIONS OF PREVIOUS PASSWORDS BEING MAINTAINED.
  AFTER   5 CONSECUTIVE UNSUCCESSFUL PASSWORD ATTEMPTS,
      A USERID WILL BE REVOKED.
  PASSWORD EXPIRATION WARNING LEVEL IS  15 DAYS.
  INSTALLATION PASSWORD SYNTAX RULES:
    RULE 1  LENGTH(8)     xxxxxxxx
   LEGEND:
    A-ALPHA C-CONSONANT L-ALPHANUM N-NUMERIC V-VOWEL W-NOVOWEL *-
ANYTHING
    c-MIXED CONSONANT m-MIXED NUMERIC v-MIXED VOWEL $-NATIONAL s-
SPECIAL
    x-MIXED ALL
```

Disabling the New Algorithm

- **Enable the new algorithm:**

  ```
  SETROPTS PASSWORD(ALGORITHM(NOALGORITHM))
  ```

- **Confirm this with SETROPTS LIST:**

```
PASSWORD PROCESSING OPTIONS:
  THE ACTIVE PASSWORD ENCRYPTION ALGORITHM IS LEGACY
  PASSWORD CHANGE INTERVAL IS  60 DAYS.
  PASSWORD MINIMUM CHANGE INTERVAL IS   3 DAYS.
  MIXED CASE PASSWORD SUPPORT IS IN EFFECT
  SPECIAL CHARACTERS ARE ALLOWED.
  10 GENERATIONS OF PREVIOUS PASSWORDS BEING MAINTAINED.
  AFTER   5 CONSECUTIVE UNSUCCESSFUL PASSWORD ATTEMPTS,
      A USERID WILL BE REVOKED.
  PASSWORD EXPIRATION WARNING LEVEL IS  15 DAYS.
  INSTALLATION PASSWORD SYNTAX RULES:
    RULE 1  LENGTH(8)     xxxxxxxx
   LEGEND:
    A-ALPHA C-CONSONANT L-ALPHANUM N-NUMERIC V-VOWEL W-NOVOWEL *-
ANYTHING
    c-MIXED CONSONANT m-MIXED NUMERIC v-MIXED VOWEL $-NATIONAL s-
SPECIAL
    x-MIXED ALL
```

Transitioning to the new Algorithm

- **KDFAES Active:**
  - New passwords will be encrypted under the active algorithm
  - Existing passwords will be evaluated regardless of their format (except for masking)
  - Same for history, which can contain entries for different formats

- **Nothing gets automatically converted to the new format:**
  - Conversion mechanism is provided:
    - Works only when existing algorithm is **DES**

      (Will otherwise create an unusable password and history)

Converting Passwords to KDFAES

- Passwords and password history which are **DES** format can be converted directly to **KDFAES**.
- When current algorithm is **KDFAES**:

```
ALTUSER USER1 PWCONVERT
```

  - Converts any legacy passwords and password history entries to KDFAES format
  - **Note:** Password phrases are not converted

- When current algorithm is **NOALGORITHM**:

```
ALTUSER USER1 PWCONVERT
```

  - This indicates that all passwords and password history entries and password phrases and password phrase history entries in **KDFAES** format should be deleted.

Password History Cleanup

- When the **SETROPTS PASSWORD(HISTORY(n))** value is changed password history entries can be stranded and never be reused.
- Previously the the RACF website provided a utility to clean these old password history values: **CUTPWHIS**

- New password history cleanup method:

```
ALTUSER USER1 PWCLEAN
```

- The new password convert option, **PWCONVERT**, will also perform a **PWCLEAN**

DBUNLOAD Password Updates

- **DBUNLOAD now displays user password fields:**
  - You can demonstrate to an auditor that passwords and password phrases are encrypted under the new algorithm with the help of new fields created by the IRRDBU00 utility.
  - A sample query is included.
- **Fields:**
  - **USBD_PWD_ALG** - Algorithm used to protect password. Values include “LEGACY”, “KDFAES”, and “NOPASSWORD”.
  - **USBD_LEG_PWDHIST_CT** - Number of legacy password history entries
  - **USBD_XPW_PWDHIST_CT** - Number of non-legacy (e.g. KDFAES) password history entries
  - **USBD_PHR_ALG** - Algorithm used to protect password phrase. Values include “LEGACY”, “KDFAES”, and “NOPHRASE”.
  - **USBD_LEG_PHRHIST_CT** - Number of legacy password phrase history entries
  - **USBD_XPW_PHRHIST_CT** - Number of non-legacy (e.g. KDFAES) password phrase history entries

Require only READ Authority for IRRDBU00 Input Data Sets if PARM=NOLOCKINPUT Specified

IRRDBU00: Require only READ Authority

- **Since its inception, IRRDBU00 has required UPDATE authority to the RACF data set(s) which are used as input**
- **With V2.2, if you specify PARM=NOLOCKINPUT, only READ authority is required**
- **Eliminates the need to use the "trick" of specifying LABEL=(,,,IN) on the input DD statement**

Certificate Enhancements

RACDCERT Granular Authority

- **Currently, the authority to issue RACDCERT commands is controlled using profiles in the FACILITY class with resources named IRR.DIGTCERT.<racdcert function>**
  - READ authority allows you to act on your own certificate or key ring
  - UPDATE allows you to act on the certificate of another user
  - CONTROL allows you to act on a CERTAUTH or SITE resource

- **There is no ability to control operations on a certificate based upon:**
  - Owner
  - Certificate label
  - Key ring name
  - Function
- **There is limited ability to allow for a segregation of RACDCERT authorities among the administrators**
- **There is no way to enforce an naming convention for certificates and key rings**

RACDCERT Granular Authority…

- **Granular control is turned on by the presence of the profile IRR.RACDCERT.GRANULAR in the RDATALIB class**

- **If the profile IRR.RACDCERT.GRANULAR does not exist, the original IRR.DIGTCERT.<racdcert function> profile(s) in the FACILITY class will be used.**

- **Applies to these 13 RACDCERT functions only:**
  - **Certificate:** ADD, ALTER, DELETE, EXPORT, GENCERT, GENREQ, IMPORT, REKEY and ROLLOVER
  - **Ring**: ADDRING and DELRING
  - **Certificate and Ring:** CONNECT and REMOVE

RACDCERT Granular Authority…

- **When granular control is enabled, one or both types of the following profiles in the RDATALIB class will be checked for READ access, depending on whether a certificate, a ring or both is involved**

- **For certificates:**
  - IRR.DIGTCERT.<cert owner>.<cert label>.UPD.<racdcert cert functions>
    - where 'cert owner' is the RACF user ID, or CERTIFAUTH (for certificate owned by CERTAUTH), or SITECERTIF (for certificate owned by SITE)
    - EXPORT may use IRR.DIGTCERT.<cert owner>.<cert label>.LST.EXPORT if no private key is exported
    - If the function involves multiple certificates, such as exporting a chain of certificates, multiple profiles will be checked

RACDCERT Granular Authority…

- **For key rings:**
  - IRR.DIGTCERT.<ring owner>.<ring name>.UPD.<ADDRING or DELRING>

- **For certificates and key rings:**
  - IRR.DIGTCERT.<cert owner>.<cert label>.LST.<CONNECT or REMOVE>
  - IRR.DIGTCERT<ring owner>.<ring name>.UPD.<CONNECT or REMOVE>

PKI Services: OCSP

- **Currently, the Online Certificate Status Protocol (OCSP) is used to get revocation status of certificates.**
  - OCSP requires server responses to be signed but does not specify a mechanism for selecting the signing algorithm to be used
- **Prior to z/OS V2.2, z/OS PKI Services could only use the same signing algorithm used for certificate and Certificate Revocation List (CRL) signing specified in the configuration file to sign the OCSP response**
- **PKI Services can now sign the OCSP response with the client specified signing algorithm through an extension in the request in the way documented by RFC6227**
- **PKI Services chooses the signing algorithm as follows:**
  - If the request contains the Preferred Signature Algorithms extension, PKI will pick the first one on the list
  - If it is not on PKI's supported list or it does not meet the contemporary standards, such as md-2WithRSAEncryption, md-5WithRSAEncryption, the next one will be used
  - If none of the specified algorithms is supported by PKI Services or meet the contemporary standard, PKI will use the one specified in the configuration file

PKI Services: Multiple Administrative Approvals

- **PKI Services supports both automatic approval mode and administrator approval mode**
- **In the administrator approval mode, only one administrator is required to approve the requests**
- **Some government agencies require all PKI products to have an “NxM” authentication factor**
  - For example, two PKI administrators have to validate a request before issuing the certificate
- **PKI Services will now allow the administrator approval mode to support multiple number of approvers**
- **A configuration option with be provided in the CGI templates file and JSP templates xml file to set the number of administrators required to approve a certificate request**
  - The option will be provided on a per template basis
- **A change of the configured number of approvers will not affect the existing certificate requests, only the new requests**

RACF Support for IBM Multifactor Authentication (IBM MFA)

z/OS Multi-Factor Authentication

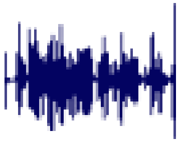
- **Multi-factor Authentication on z/OS** provides a way to raise the assurance level of OS and applications / hosting environments by extending RACF to authenticate users with multiple authentication factors

000147678169 Options

Tokencode:
6760 2680

Re-enter PIN Copy

RSA SecurID



- **Authentication Factors:**
  - Something you know
    A password / PIN Code
  - Something you have
    ID badge or a cryptographic key
  - Something you are
    Fingerprint or other biometric data

- **Today on z/OS, users can authentication with:**
  - Passwords, Password phrases, PassTickets, Digital Certificates, or via Kerberos

- **Today's problem:**
  - 2014 Verizon Data Breach Investigations Report said 2 out of 3 breaches involved attackers using stolen or misused credentials.
  - In the case of an attempted breach using comprised credentials, the extra protection that MFA provides can make the difference between having a secured vs. compromised system.
  - Breaches impact clients financially, their customers, and their reputations

IBM *Multi-Factor Authentication for z/OS*

Higher assurance authentication for IBM z/OS systems that use RACF

- **IBM Multi-Factor Authentication on z/OS** provides a way to raise the assurance level of OS and applications / hosting environments by extending RACF to authenticate individual users:
- **Support for third-party authentication systems**
  - RSA® Ready supporting RSA SecurID® Tokens (hardware & software based)
  - Direction to support IBM TouchToken – Timed One time use Password (TOTP) generator token
  - Direction to support PIV/CAC cards - Commonly used to authenticate in the Public Sector enterprises
- **Tightly integrated with SAF & RACF**
  - RACF provides the configuration point to describe multi-factor authentication requirements down to a per User ID basis
  - Deep RACF integration for configuration and provisioning data stored in RACF database allowing seamless back-up and recovery

Typical Client Use Cases:

- **Enable higher-security user logins** on IBM z/OS systems that use RACF for security
- Enable strong authentication for employees that carry **iOS devices** or **RSA SecurID** tokens

TECHNOLOGY RSA READY PARTNER

Fast, flexible, deeply integrated, easy to deploy, easy to manage, and easy to use.

Achieve regulatory compliance, reduce risk to critical applications and data

Architecture supports multiple third-party authentication systems at the same time

RACF & MFA Services and Related Support

- **RACF MFA support introduces extensions to a variety of components of RACF**
  - User related commands
    - Allow the provisioning and definition of the acceptable MFA tokens for a user
    - Definition of **authentication token types**
  - **Extensions to SAF programming interfaces**
    - Provides new SAF services for z/OS MFA Services allowing the access to MFA data stored in the RACF database.
  - **Auditing extensions**
    - Tracks which factors used during the authentication process for a given user
  - **Utilities**
    - RACF Database unload non-sensitive fields added to the RACF database used by MFA processing
    - SMF Unload – unloads additional relocate sections added to SMF records
      - Related to the tokens used on a specific authentication event
- **z/OS MFA Services started task**
  - z/OS MFA address space which tracks state for user authentication events
  - Provides an anchor for communications for factors such as RSA SecurID

MFA RACF User Profile Management

- **MFA Factor fields is stored in the RACF user profile**
- **Defined by a RACF Administrator via ALTUSER command**

- **Example ALTUSER Syntax:**

```
      [ MFA(
          [ PWFALLBACK | NOPWFALLBACK ]
          [ FACTOR(factor-name) | DELFACTOR(factor-name) ]
          [ ACTIVE | NOACTIVE ]
          [ TAGS(tag-name:value …) ]
              | DELTAGS(tag-name … )
              | NOTAGS ]
      )
      | NOMFA ]
```

- **RACF will call the MFA Services task to validate the factor specific information that is specified on the ALTUSER command TAGS keyword**
  - If a syntax error or unknown name value pair is supplied MFA Services will reflect an error to RACF
    - RACF issues a message and a MFA Services provided message which indicates the nature of the syntax error

RACF Health Check Updates

New and Updated RACF Health Checks

- RACF is introducing four new health checks
  - **RACF_CSFKEYS_ACTIVE**
  - **RACF_CSFSERV_ACTIVE**
  - **RACF_PASSWORD_CONTROLS**
  - **RACF_ENCRYPTION_ALGORITHM**

- RACF is updating the **RACF_SENSITIVE_RESOURCE** to:
  - Report on the protection status of ICSF TKDS, PKDS, and CKDS data sets
  - Report on the protection status of the RACF remote sharing (RRSF) work data sets
  - Report on the protection status of additional z/OS UNIX resources

New and Updated RACF Health Checks

- **RACF_CSFKEYS_ACTIVE** and **RACF_CSFSERV_ACTIVE** raise an exception if the class is inactive

```
CHECK(IBMRACF,RACF_CSFKEYS_ACTIVE)
SYSPLEX:    LOCAL     SYSTEM: RACFR22
START TIME: 03/05/2014 16:45:04.542092
CHECK DATE: 20140106  CHECK SEVERITY: MEDIUM
CHECK PARM: CSFKEYS


* Medium Severity Exception *

IRRH229E The class CSFKEYS is not active.

  Explanation:  The class is not active. IBM recommends that the
    security administrator evaluate the need for this class, define
    profiles in it as appropriate, and activate the class.

. . .
. . .

  Automation:  None.

  Check Reason:  IBM recommends activating this class

END TIME: 03/05/2014 16:45:04.606623  STATUS: EXCEPTION-MED
```

New and Updated RACF Health Checks

- **RACF_SENSITIVE_RESOURCES** is updated to examine the ICSF CKDS, PKDS, and TKDS VSAM data sets

```
CHECK(IBMRACF,RACF_SENSITIVE_RESOURCES)
SYSPLEX:    LOCAL     SYSTEM: RACFR22
START TIME: 03/05/2014 17:52:23.177975
CHECK DATE: 20120106  CHECK SEVERITY: HIGH
. . .
. . .
. . .


                          ICSF Dataset Report

S Data Set Name                           Vol    UACC Warn ID*  User
- --------------------------------------- ------ ---- ---- ---- ----
  RACFDRVR.ICSF.PKDS                      D94001 None No   ****
  RACFDRVR.ICSF.CKDS                      D94001 None No   ****
  RACFDRVR.ICSF.TKDS                      D94001 None No   ****

. . .
. . .
. . .
```

New and Updated RACF Health Checks

- **Notes on these Checks:**
  - Clients who are not using ICSF may chose to make the ICSF checks INACTIVE using a HZSPRMxx PARMLIB statement.
  - **RACF_CSFSERV_ACTIVE**, **RACF_CSFKEYS_ACTIVE**, and the ICSF CKDS, PKDS, and TKDS update to **RACF_SENSITIVE_RESOURCES** is being shipped on releases V1.12, V1.13, and V2.1 with APAR OA44696.

New and Updated RACF Health Checks

- **RACF_PASSWORD_CONTROLS** examines basic password controls
- Clients can modify the IBM recommendation with a health check parameter

```
CHECK(IBMRACF,RACF_PASSWORD_CONTROLS)
SYSPLEX:    LOCAL     SYSTEM: RACFR22
START TIME: 03/05/2014 16:45:04.494234
CHECK DATE: 20140118  CHECK SEVERITY: MEDIUM

                         RACF Password Controls

S Control                                           Value Target
- ------------------------------------------------- ----- ------
E Mixed case passwords are allowed                   NO   YES
  Number of consecutive unsuccessful logon attempts   3     3

* Medium Severity Exception *

IRRH283E The RACF_PASSWORD_CONTROLS check found an exception
with one or more password control settings.
. . .
. . .
  Automation:  None.

  Check Reason:  Password control recommendations should be used.

END TIME: 03/05/2014 16:45:04.603476  STATUS: EXCEPTION-MED
```

New and Updated RACF Health Checks

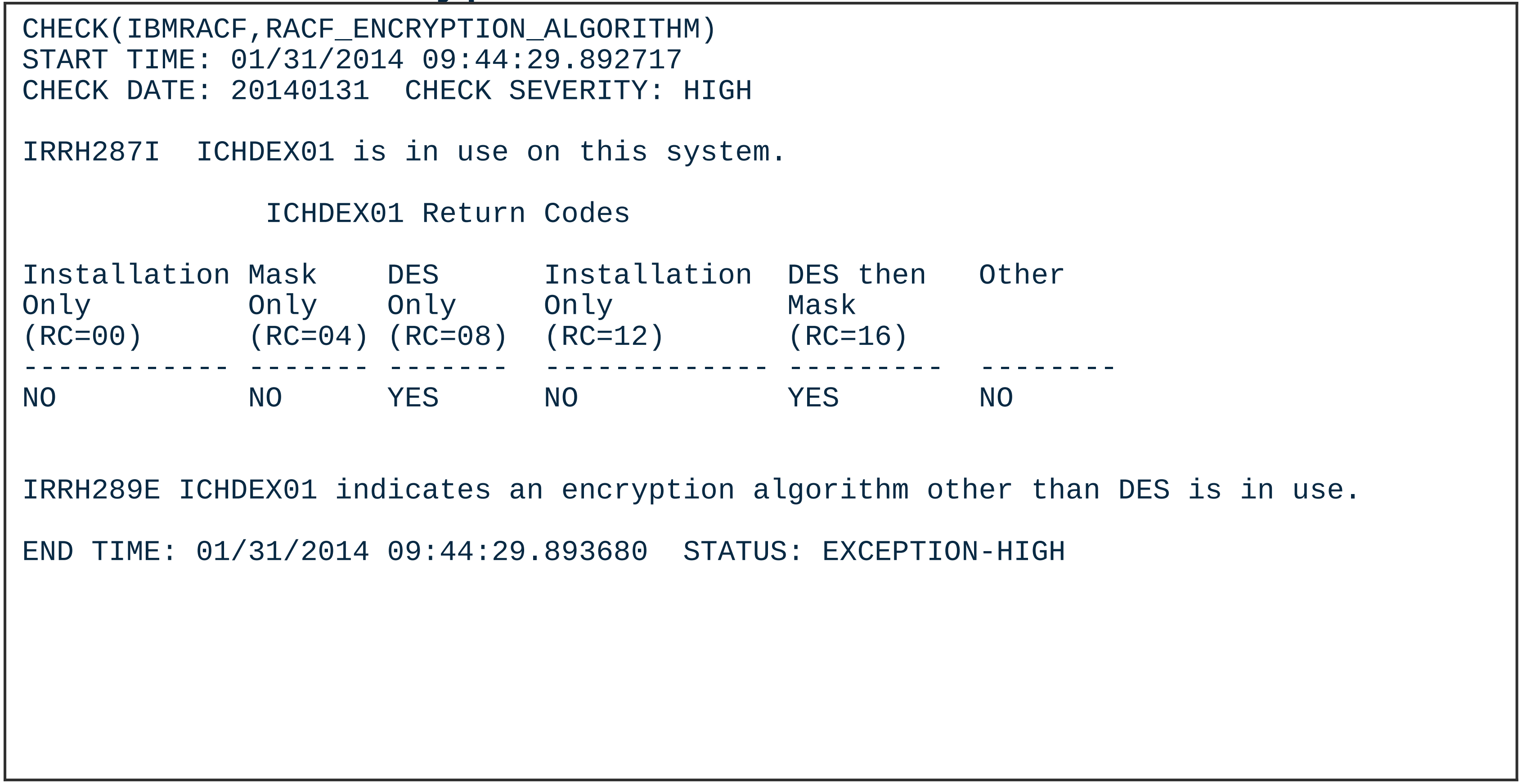
- **RACF_ENCRYPTION_ALGORITHM** examines the return code from the RACF encryption exit (ICHDEX01) exit for authentication and raises an exception if anything other than DES (or something stronger in the future) is in use.

- **ICHDEX01** communicates the desired encryption with a return code:
  - 00: Installation-defined
  - 04: Masking
  - 08: DES
  - 12: Installation-defined
  - 16: DES then masking

- The lack of **ICHDEX01** currently requests DES then the RACF “masking” algorithm and is considered an exception.

New and Updated RACF Health Checks

- The **RACF_ENCRYPTION_ALGORITHM** check reports on the use of encryption for authentication since the last IPL



```
CHECK(IBMRACF,RACF_ENCRYPTION_ALGORITHM)
START TIME: 01/31/2014 09:44:29.892717
CHECK DATE: 20140131  CHECK SEVERITY: HIGH

IRRH287I  ICHDEX01 is in use on this system.

              ICHDEX01 Return Codes

Installation Mask    DES      Installation  DES then   Other
Only         Only    Only     Only          Mask
(RC=00)      (RC=04) (RC=08)  (RC=12)       (RC=16)
------------- ------- -------  ------------- ---------  --------
NO           NO      YES      NO            YES        NO


IRRH289E ICHDEX01 indicates an encryption algorithm other than DES is in use.

END TIME: 01/31/2014 09:44:29.893680  STATUS: EXCEPTION-HIGH
```

New and Updated RACF Health Checks

- **Notes on RACF_PASSWORD_CONTROLS and RACF_ENCRYPTION_ALGORITHM:**
  - These checks are being shipped on releases V1.12, V1.13, and V2.1 with an APAR to be determined later.

New and Updated RACF Health Checks

- **RACF_SENSITIVE_RESOURCES** is updated to examine the RRSF input and output data sets

```
CHECK(IBMRACF,RACF_SENSITIVE_RESOURCES)
SYSPLEX:    LOCAL     SYSTEM: RACFR22
START TIME: 03/05/2014 17:52:23.177975
CHECK DATE: 20120106  CHECK SEVERITY: HIGH
. . .
. . .
. . .


                          RRSF Dataset Report

S Data Set Name                            Vol    UACC Warn ID*  User
- ---------------------------------------- ------ ---- ---- ---- ----
  SYS1.MVSX.INMSG                          D94001 None No   ****
  SYS1.MVSX.MVSA.INMSG                     D94001 None No   ****
  SYS1.MVSX.MVSA.OUTMSG                    D94001 None No   ****
  SYS1.MVSX.MVSB.INMSG                     D94001 None No   ****
  SYS1.MVSX.MVSB.OUTMSG                    D94001 None No   ****
. . .
. . .
. . .
```

New and Updated RACF Health Checks

- The **RACF_SENSITIVE_RESOURCES** check is updated to check on these z/OS UNIX resources:
  - **FACILITY class:**
    - BPX.POE
    - BPX.JOBNAME
    - BPX.FILEATTR.SHARELIB
    - BPX.SMF
    - BPX.STOR.SWAP
    - BPX.UNLIMITED.OUTPUT
  - **UNIXPRIV class:**
    - SUPERUSER.FILESYS.QUIESCE
    - SUPERUSER.FILESYS.PFSCTL
    - SUPERUSER.FILESYS.VREGISTER
    - SUPERUSER.IPC.RMID
    - SUPERUSER.SETPRIORITY
  - **SURROGAT class:**
    - BPX.SRV.<userid>

New RACF Health Checks

- **APAR OA45608 for V1.12(UA74753), V1.13 (UA74754), V2.1 (UA74755) introduces these two new health checks:**
  - **RACF_ENCRYPTION_ALGORITHM**, which raises an exception if “weak” (less 'secure' than DES) encryption is allowed for logon passwords
    - Having no ICHDEX01 is considered an exception as the absence of ICHDEX01 allows masked passwords
  - **RACF_PASSWORD_CONTROLS**, which raises an exception if:
    - Mixed case passwords are not in effect or
    - The maximum number of consecutive failed logon attempts is greater than 3 or
    - A password/password phrase can be valid for more than 90 days

- **APAR OA44496 V1.12(UA73744), V1.13 (UA73745), V2.1 (UA73746) introduces these two new checks:**
  - **RACF_CSFSERV_ACTIVE,** which raises an exception if the CSFSERV class is not active
  - **RACF_CSFKEYS_ACTIVE,** which raises an exception if the CSFKEYS class is not active
  - … and adds checks for the ICSF CKDS, PKDS, and TKDS data sets to the **RACF_SENSITIVE_RESOURCES** health check

New RACF Health Checks…

- **APAR OA48714 for V1.13 (UA81588), V2.1 (UA81589), and V2.2 (UA81587) introduces these two new health checks:**

  - **RACF_JESJOBS_ACTIVE,** which raises an exception if the JESJOBS class is not active
  - **RACF_JESSPOOL_ACTIVE,** which raises an exception if the JESSPOOL class is not active
  - **RACF_BATCHALLRACF,** which raises an exception if SETROPTS JES(BATCHALLRACF)) is not in effect

Signed SMF Records

Signed SMF Records

- **With z/OS V2.2, you can configure SMF to sign SMF records**
  - Idea is to make SMF a fully-trusted repository of audit data by making it much more tamper-evident
  - Available for SMF data written to the System Logger
  - Uses both CPACF symmetric algorithm for hashing to support needed data rates and CEXnC card for signatures
  - Groups of records are signed
  - Each group will have a new SMF type 2 trailer record with the signature
  - IFASMFDP support planned for verifying the signatures
  - To verify signatures:
    - Unload using IFASMFDL
    - Process the SMF data with IFASMFDP
  - The SMF type 2 record format is documented, so anyone can do signature verification

New DFSMSdfp Function

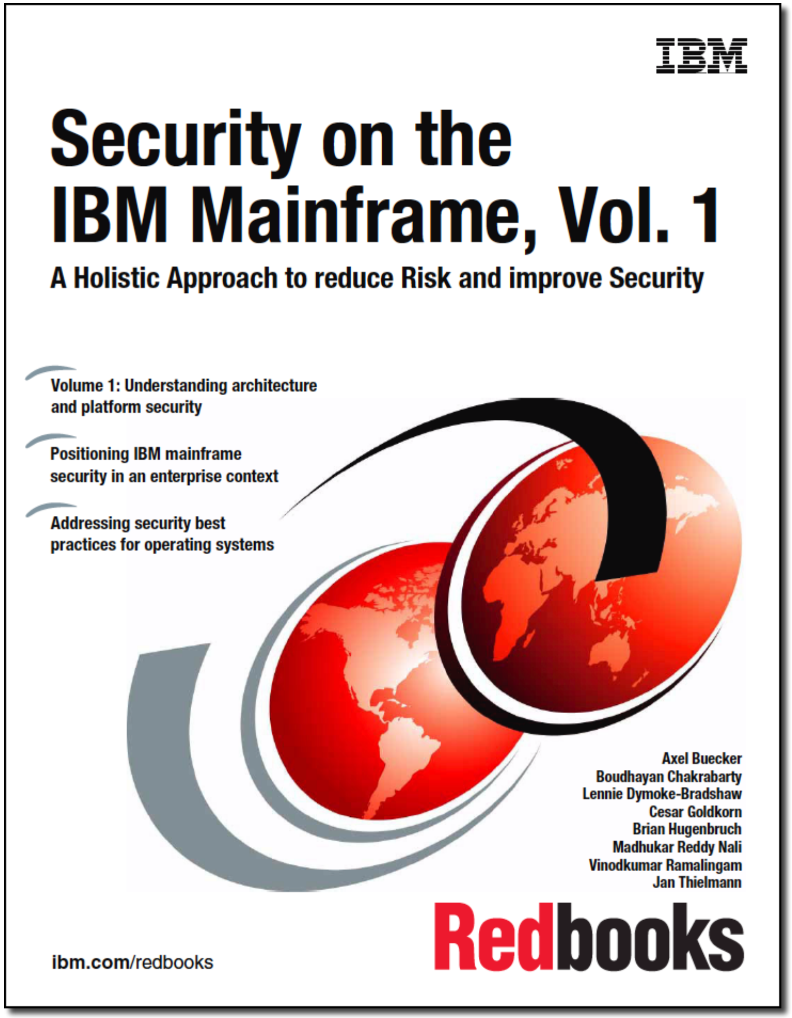
New DFSMSdfp Function: BYPASS_AUTH

- **The program properties table (PPT) allows an installation to define those programs which, when executed as the job step program in the first step of a started task, bypass the calls to the external security manager when a data set is opened.**
  - Programs run in this manner will have the JSCBPASS bit in the job step control block (JSCB) set on.
  - JSCBPASS applies to all data set accesses
  - ACEEPRIV performs a similar function
- **Authorized programs can set this bit on and thus bypass data set access controls**
- **With APAR OA48124 V1.13 (UA81032), V2.1(UA81033), and V2.2 (UA81034) , DFSMSdfp is offering an alternative mechanism for authorized code to bypass the data set authorization check**
  - **A new option on the DCBE macro (BYPASS_AUTH) that bypasses DFP's authorization check if the invoker is authorized (APF, system key, or supervisor state) at the time of the OPEN**
  - **New fields in the existing SMF 14/15 records indicate:**
    - Was BYPASS=AUTH specified (SMF14RFG1DBYP)?
    - Was JSCBPASS ON at the time of the open(SMF14RFG1JBYP)?
    - Was the caller in supervisor state, system key, APF-authorized (SMF14RFG1AUTH)?
    - Did DFP bypass the authorization call (SMF14RFG1BYP)?
- **Changing authorized applications to use this BYPASS_AUTH helps implement the "principle of least privilege" for the application**

Updated DFSMSdfp Function: Erase-on-Scratch Improvements

- **DFSMSdfp has made significant improvements in Erase-on-Scratch performance in z/OS V2.1 and z/OS V2.2.**
  - Up to 255 tracks erased in a single channel program in z/OS V2.1
  - Up to 12,240 tracks in a single channel program in z/OS V2.2
- **Frank Kyne performed erase-on-scratch testing that is documented in Cheryl Watson's "TUNING Letter - 2015 No. 1":**
  - Allocated data sets of 1, 100, 255, 25600, and 63000 tracks
  - Ran a separate job to delete each data set, varying erase-on-scratch on and off, on z/OS V1.13 and z/OS V2.1
- **Frank's results:**
  - Small reduction in elapsed time and EXCP counts for the smaller data set sizes (1, 100, 255)
  - Large reduction in elapsed time and EXCP counts for the larger data sets
    - For the 63,000 track data set, EXCPs dropped from 63,007 to 263
    - Elapsed times decrease between 1/3 and 2/3
- **Remember that z/OS V2.2 increases the upper limit on the number of tracks erased in a single CCW to 12,240 (from 255)**

Helpful Publications



Security on the IBM Mainframe, Vol. 1

A Holistic Approach to reduce Risk and improve Security

Volume 1: Understanding architecture and platform security

Positioning IBM mainframe security in an enterprise context

Addressing security best practices for operating systems

Axel Buecker
Boudhayan Chakrabarty
Lennie Dymoke-Bradshaw
Cesar Goldkorn
Brian Hugenbruch
Madhukar Reddy Nali
Vinodkumar Ramalingam
Jan Thielmann

ibm.com/redbooks

Redbooks

Helpful Publications…

- SA23-2290 - z/OS Security Server RACF Callable Services
- SA23-2292 - z/OS Security Server RACF Command Language Reference
- GA32-0885 - z/OS Security Server RACF Data Areas
- SA23-2288 - z/OS Security Server RACF Macros and Interfaces
- SA23-2291 - z/OS Security Server RACF Messages and Codes
- SA23-2289 - z/OS Security Server RACF Security Administrator's Guide
- SA23-2287 - z/OS Security Server RACF System Programmer's Guide
- SA23-2294 - z/OS Security Server RACROUTE Macro Reference
- GA32-0886 - z/OS Security Server RACF Diagnosis Guide
- SA23-2286 - z/OS Cryptographic Services PKI Services Guide and Reference
- SC14-7495 - z/OS Cryptographic Services System Secure Sockets Layer Programming
- SA23-2231 - z/OS ICSF Writing PKCS #11 Applications
- SA23-2284 - z/OS UNIX System Services: Messages and Codes
- SA23-2281 - z/OS UNIX System Services Programming: Assembler Callable Services Reference
- SC27-3651 - z/OS Communication Server: IP Configuration Guide
- GC27-2652 - z/OS Communication Server: IP Diagnosis Guide
- SC27-3661 - z/OS Communication Server: IP System Administrator's Commands
- SA23-6843 - IBM Health Checker for z/OS User's Guide

RACF® Update