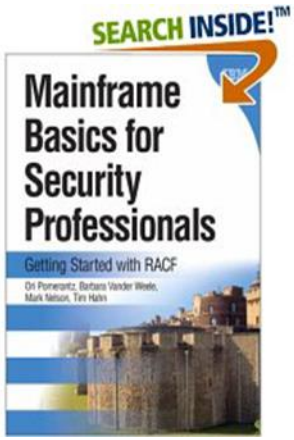
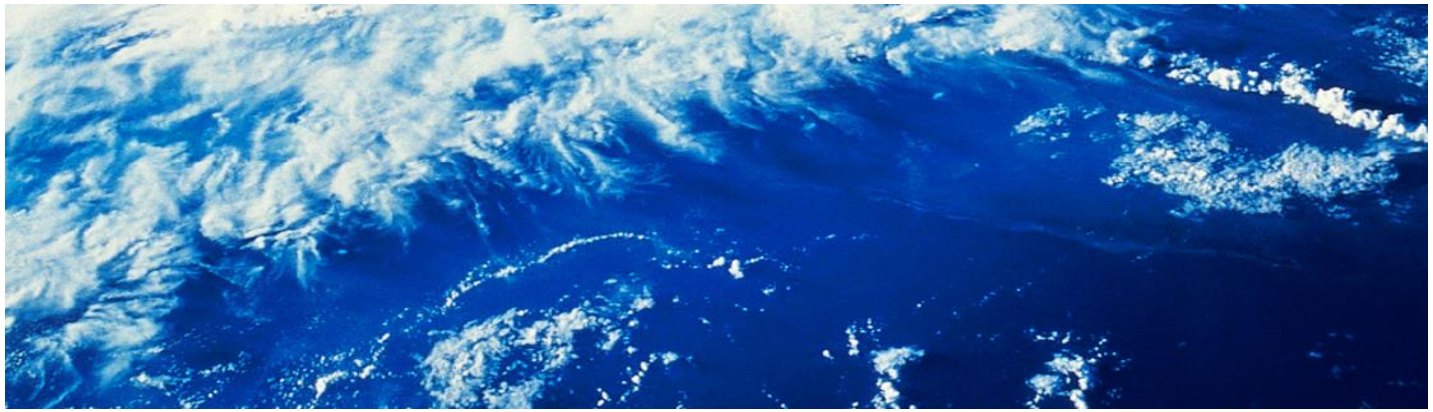
RACF® Update for z/OS® V2.2

New York RACF Users Group / Tampa RACF Users Group
20 November 2015

Mark Nelson, CISSP®, CSSLP®
RACF Design and Development
IBM Poughkeepsie
markan@us.ibm.com





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
- **Pre-V2.2 Enhancements**
  - New Health Checks

Common Criteria Update

Common Criteria Update

- **Recent Common Criteria Evaluations of Interest:**
  - z/OS V2.1, OSPP 3.9, August, 2015
  - 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 EC12 GA2/BC12 GA1 EAL5+, 19 February, 2014
  - PR/SM for IBM zEnterprise EC12 GA1 EAL5+, 19 February, 2013
- 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.**
4) 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.
5) 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.**
6) **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.
7) 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.
8) On the new main system, issue TARGET OPERATIVE commands to make the connection with itself and all connections with remote nodes operative.
9) **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.
10) **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.
11) 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 V2R2 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

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 V2R2

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**

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:**

  <ring owner>.<ring name>.UPD.<ADDRING or DELRING>

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

Health Check Updates

New RACF Health Checks

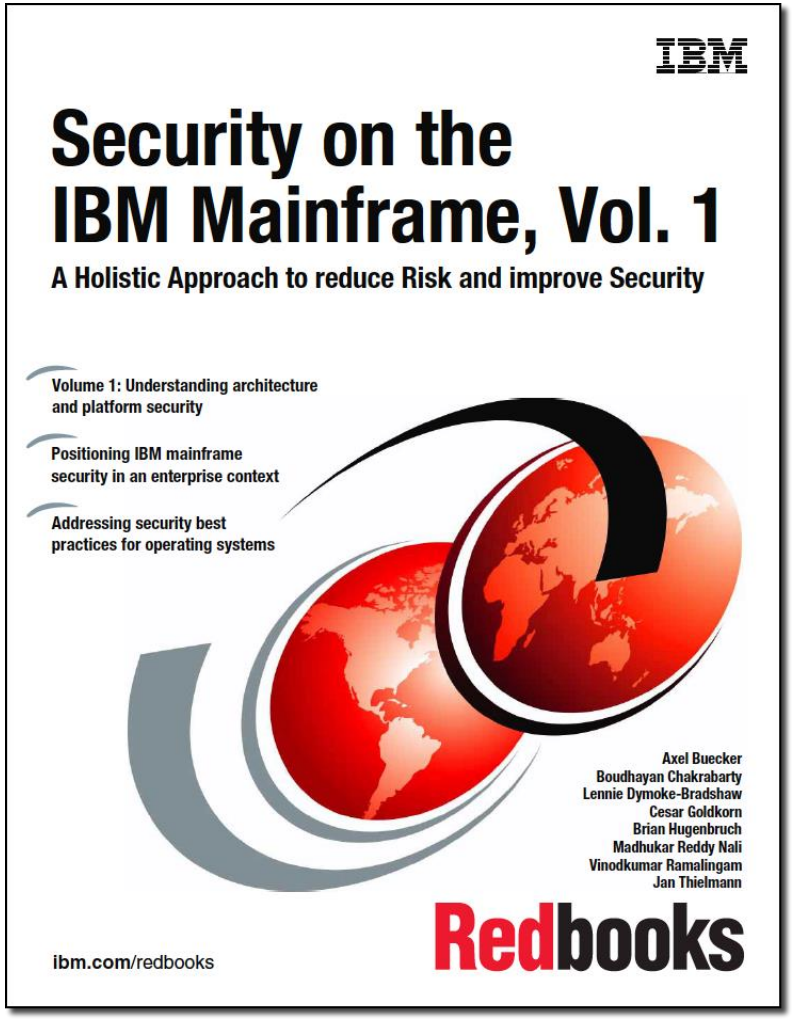
- **APAR OA45608 for V1.12(UA74753), V1.13 (UA74754), V2.1 (UA74755) introduces these two new health checks:**
  - **RACF_ENCRYPTION_ALGORITHM**, which raises an exception if “weak” (less 'secure' than DES) encryption is allowed for logon passwords
    - Having no ICHDEX01 is considered an exception 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

Shameless Plug: Hot Topics #29: August, 2015

- Don't Fall on your p@sSword
- Secure, but not foolproof
- Your order's up! RACF client requirements satisfied in z/OS V2.2
- Erasure and encryption: The yin and yang of security technologies
- Drowning in digital certificates? Here's a lifeline!
- Give credit to Crypto; It gives Crypto to Credit
- Fortify your SMF data with digital signatures

Available at http://www.ibm.com/systems/z/os/zos/library/hot-topics/hot-topics.html

Helpful Publications



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