RACF® Update

SHARE Orlando
Session 17550 - August 11 2015

Eric Rosenfeld, CISSP®
Security Design and Development
IBM Poughkeepsie
rosenfel@us.ibm.com

Agenda

- **Password Updates**
- **A Look at the Future – V2R2**
  - **Additional Password Updates**
  - **Read-Only Auditor**
  - **Digital Certificates**
  - **PKI**
  - **UNIX Search**
  - **RRSF**
    - **Dynamic Main**
    - **Unidirectional Connections**

z/OS V2.1 RACF Statement of General Direction

- **Enhanced RACF password encryption algorithm:**
  - In the future, an enhanced RACF password encryption algorithm is planned. This support will be designed to provide improved cryptographic strength in RACF password algorithm processing. This will be intended to help protect RACF password data in the event that a copy of a RACF database becomes inadvertently accessible.
- The future is ***now!***

Background

- **Since its first release in 1976, RACF has supported the password as a primary authentication mechanism**
  - Originally, passwords were stored in a “masked” format
  - Reversible!
- **With RACF 1.6 (1984) RACF introduced a the “Data Encryption Standard” (DES) as an option for the storage of passwords**
  - Value stored in the RACF database is the user ID encrypted with the password
  - Not reversible, other than by “brute force”
- **The encryption algorithm was selected using an exit, ICHDEX01, located in LPA**
  - Return code 04: Use masking algorithm
  - Return code 08: Use DES
  - Return code 16: Use DES, fall-back to masking
  - No exit: Use DES than masking

Background…

- **Ask yourself this question: "Which is a better encryption algorithm?" Your possible answers are:**
  - DES
  - AES
  - The question contains insufficient information to allow for a correct answer
- **The most important element in the question isn't the algorithm... it's the size and character set of the key!**
  - And what's the size of the key? It's the 8-byte password!
  - You can make the key space larger by enabling mixed-case passwords
- **Password phrases are a marvelous mechanism for resilience against brute force attacks**
  - Wouldn't it be nice if you could have password phrase only users?
- **Resilience against brute-force password attacks is affected by**
  - The size and non-predictability of the key
  - The speed of the algorithm (Faster isn't better!)

The Paradox

- **Why does slowing down the encryption process help against a brute-force attack?**
  - You only have to do the algorithm once for a password validation.
  - The attacker has to do the algorithm once for each brute force attempt The number of brute-force attempts needed is a function of the size of the key, the character set of the key…. and luck
  - **Net:** You are slowed down a little... the attacker is slowed down ***a lot!***

Other Considerations and Concerns

- **RACF's password processing is very well known**
  - Some resource managers perform their processing knowing what RACF's processing is
  - Some extract the cipher text password and then perform their own validation
  - Some present a ciphertext value during the authentication process
  - Some compute the ciphertext password themselves and insert that into the user profile
- **The challenge is to get all of these to work with what RACF implements**
  - Some vendor applications will have to change
- **Enablement must be optional**

New Password Processing in RACF

- **New function APARs OA43998 (SAF)/OA43999(RACF)**
  - Migrate from 56-bit single key DES to key-derived AES (KDFAES)
  - Password-phrase-only users
  - Password expiration by administrator action
  - Password history cleanup
  - Additional “special” characters allowed in passwords
  - Rolled back to z/OS V1.12
- **A number of products are effected by these enhancements**
- **New SMP/E FIXCATEGORIES are defined for each function so that you can identify updates as they become available**
  - IBM.Function.RACF.PasswordCharacters
  - IBM.Function.RACF.PasswordEncryption
- **Informational APAR II14765 documents known restrictions**

KDFAES

- **With KDFAES (key derivation function with AES), the password or password phrase is appended with random data, then is iteratively hashed thousands of times to derive a 256-bit encryption key. That key is used to AES encrypt the user ID which has been appended with other data.**
- **Enabling the new encryption processing is done with the SETROPTS command**
  - `SETROPTS PASSWORD(ALGORITHM(KDFAES))`
  - New passwords will be encrypted using the new algorithm
- **You can convert a user's password and password history to KDFAES using the new ALTUSER PWCONVERT keyword:**
  - `ALTUSER userID PWCONVERT`
  - You can use a simple SEARCH command to create the commands to convert all users to KDFAES

Other Password/Password Phrase Enhancements

- **A password phrase may now be assigned to a user without requiring a password**
  - `ALTUSER userID NOPASSWORD`
- **A user's password and password phrase may now be expired without having the administrator change them**
  - `ALTUSER userID EXPIRED`
- **A user's password and password phrase history can be "cleaned up" of orphaned entries caused by the lowering of the SETROPTS PASSWORD(HISTORY(nn)) value**
  - `ALTUSER userID PWCLEAN`
- **With KDFAES active, RACF allows a password phrase of 9-13 characters without having an ICHPWX11 exit being active**

New Special Characters

- **New special characters are enabled with the SETROPTS command**
  - `SETROPTS PASSWORD(SPECIALCHARS)`
- **Two new values are available for your SETROPTS password rules:**
  - **SPECIAL**
    - Includes all of the new special characters plus the national characters '#'(X'7B'), '$' (X'5B') and "@" (X'7C')
  - **MIXEDALL**
    - Allows all password characters
    - Can be used to force selections from each character grouping (upper case, lower case, numeric, and national/special) depending on the number of MIXEDALL positions and SETROPTS MIXEDCASE is in effect

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

Related Enhancements

- **RACF Database Unload Utility (IRRDBU00) User Basic Data (0200) record updated to contain:**
  - The algorithm used to protect the password for the user
  - The algorithm used to protect the password phrase for the user
  - Legacy password history count
  - Legacy password phrase history count
  - KDFAES password history count
  - KDFAES password phrase history count
- **RACF SMF Unload Utility (IRRADU00)**
  - New keywords unloaded for ALTUSER, SETROPTS
  - RACF SMF type 81 initialization record new fields for SPECIALCHARS and encryption algorithm information

Related Enhancements…

- **With APAR OA44696, RACF has provided a new health check, RACF_ENCRYPTION_ALGORITHM**
- **RACF_ENCRYPTION_ALGORITHM 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 allow masked passwords
- **Sample Check output when ICHDEX01 is absent and KDFAES is not enabled**

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

IRRH295E  The RACF_ENCRYPTION_ALGORITHM check has detected an
exception. ICHDEX01 is not in use on this system. DES encryption
falls back to RACF masking.

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


Related Enhancements…

- **Sample Check output when ICHDEX01 is present with RC=8 (DES) only and KDFAES is not enabled**

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

IRRH296I  ICHDEX01 is in use on this system.

              ICHDEX01 Return Codes

Installation Mask    DES      Installation  DES then  Other
Only         Only    Only     Only          Mask
(RC=0)       (RC=04) (RC=08)  (RC=12)       (RC=16)   (RC=OTHER)
------------ ------- -------  ------------- --------- ----------
NO           NO      YES      NO            NO        NO


IRRH297I ICHDEX01 indicates that only DES encryption is in use.

IRRH299I No exceptions are detected.

END TIME: 01/31/2014 09:44:29.893680  STATUS: SUCCESSFUL
```

Related Enhancements…

- **Sample Check output when ICHDEX01 is present with non RC=8 (DES) and KDFAES is not enabled**

```
CHECK(IBMRACF,RACF_ENCRYPTION_ALGORITHM)
SYSPLEX:    LOCAL     SYSTEM: RACFR21
START TIME: 02/20/2015 16:32:23.153585
CHECK DATE: 20140131  CHECK SEVERITY: MEDIUM

IRRH296I 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)  (RC=OTHER)
------------ ------- ------- ------------ -------- ----------
NO           YES     NO      NO           NO       NO

 * Medium Severity Exception *

IRRH298E ICHDEX01 indicates that an algorithm other than DES
encryption is in use.

  Explanation:  The RACF_ENCRYPTION_ALGORITHM check verifies that only
    the KDFAES or DES encryption algorithm is used for password
    protection. The ICHDEX01 exit indicates the algorithm to use for
    password protection when KDFAES is not enabled. ICHDEX01 has set a
    return code indicating to use an algorithm other than DES which
    raises an exception.

    See the z/OS Security Server RACF System Programmer's Guide for more
    information about the ICHDEX01 exit.
```


Related Enhancements…

- **With KDFAES enabled, ICHDEX01 (if present) is used for the password history only**
  - **RACF_ENCRYPTION_ALGORYTHM does not raise an exception**
  - **ICHDEX01 return code information is displayed**

```
CHECK(IBMRACF,RACF_ENCRYPTION_ALGORITHM)
SYSPLEX:    LOCAL     SYSTEM: RACFR21
START TIME: 02/20/2015 16:36:05.414771
CHECK DATE: 20140131  CHECK SEVERITY: MEDIUM

IRRH294I KDFAES encryption is enabled on this system. If present,
ICHDEX01 is used only for password history.

IRRH296I ICHDEX01 is in use on this system.


                        ICHDEX01 Return Codes

Installation DES     DES     Installation DES      DES
Only         Only    Only    Only         Only     Only
(RC=00)      (RC=04) (RC=08) (RC=12)      (RC=16)  (RC=OTHER)
------------ ------- ------- ------------ -------- ----------
NO           YES     NO      NO           NO       NO

IRRH299I No exceptions are detected.

END TIME: 02/20/2015 16:36:05.415144  STATUS: SUCCESSFUL
```

Related Enhancements…

- **RACF_PASSWORD_CONTROLS 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
- **Sample RACF_PASSWORD_CONTROLS output:**

```
CHECK(IBMRACF,RACF_PASSWORD_CONTROLS)
SYSPLEX:    LOCAL     SYSTEM: RACFR21
START TIME: 09/08/2014 10:18:11.430293
CHECK DATE: 20140118  CHECK SEVERITY: MEDIUM
CHECK PARM: REVOKE(3),MIXEDCASE(YES),INTERVAL(90)


                        RACF Password Controls

S Control                                          Value Target
- ------------------------------------------------ ----- -------
E Mixed case passwords are allowed                  NO    YES
E Maximum number of consecutive failed logon attempts None  003
  Maximum days a password/passphrase is valid       030   090

* Medium Severity Exception *
IRRH283E The RACF_PASSWORD_CONTROLS check found an exception
with one or more password control settings.

  Explanation:  The RACF_PASSWORD_CONTROLS check lists each password
    control setting that is checked. Only those password control
    settings that do not meet the specified target result in an
    exception. The password control checks that result in an exception
    have an an "E" (Exception) in the "S" (Status) column.
```


Implementation Considerations

- **Before activating KDFAES or SPECIALCHARS, be sure to:**
  - Apply the OA43998/OA43999 PTFs on all systems sharing the RACF DB
  - Apply service to any products which are impacted by this new support
  - Verify that you have no “home grown” code which is affected
  - Determine the impact to your RACF exits (such as ICHDEX01/ICHPWX11)
  - Determine the impact to RACF “downloads” that you might use
  - Ensure that you have sufficient space in your RACF database to support the expansion of user profiles
  - For better performance, ensure that you are running on a processor which has the Central Processor Assist for Cryptographic Function (CPACF) to perform the SHA-256 operations.
  - Ensure that you are using ACEE caching in VLF (IRRACEE VLF class)
  - Ensure that your RRSF systems have OA43998/OA43999 applied and have consistent password settings
- **After activation, be sure to:**
  - Monitor your RACF DB for fragmentation and storage utilization
    - IRRUT200 utility

z/OS V2R2

More Password Enhancements

- **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**

Read-Only Auditor

- Allow a user to be defined (or altered) so that the user can list all information about any RACF profile without needing to grant that user additional authority to those profiles.
- User is unable to set auditing controls on profiles, but may view information in them.
- Similar to, but distinct from, the existing AUDITOR attribute – does not include the AUDITOR attribute's ability to control RACF profiles.
- Suitable for use by an external auditor who may need to verify the current security state of a system

Read-Only Auditor

- ADDUSER and ALTUSER
  - New Keywords:
    ROAUDIT - Set Read-Only Auditor
    NOROAUDIT - Remove Read-Only Auditor
- "List" commands updated to honor ROAUDIT
  LISTDSD, LISTGRP, LISTUSER, RLIST, SETROPTS LIST, SEARCH
- z/OS UNIX ck_access honors ROAUDIT
- Utilities honor ROAUDIT

  DSMON, IRRUT100, IRRXUTIL
- ROAUDIT is distinct from the existing AUDITOR attribute.
  - If ROAUDIT and AUDITOR are set, AUDITOR attribute takes precedence

Digital Certificates

- Certificate and key ring administration in RACF is handled by the RACDCERT command
- RACDCERT functions access is controlled by the FACILITY class, through the profiles IRR.DIGTCERT.<racdcert function>
- The access needed is based on the ownership of the certificates or key rings
  - READ to act on your own
  - UPDATE to act on other's
  - CONTROL to act on CERTAUTH / SITE
  - This access model is either 'none' or 'all', no granular control

Digital Certificates

- Provide RACDCERT granular control based on
  - owner
  - certificate label
  - key ring name
  - Function
- Enable segregation of RACDCERT authorities among the administrators
- Enforce a naming convention for naming the certificates and keyrings

Digital Certificates

- 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

Cert	Ring	Cert and Ring
ADD	ADDRING	CONNECT
ALTER	DELRING	REMOVE
DELETE		
EXPORT		
GENCERT		
GENREQ		
IMPORT		
REKEY		
ROLLOVER		

Digital Certificates

- When granular control is turned on, 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, eg, exporting a chain of certificates, multiple profiles will be checked

Digital Certificates

- 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 - OCSP

- RFC 2560, 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
- z/OS PKI Services uses the same signing algorithm used for certificate and Certificate Revocation List (CRL) signing specified in the configuration file to sign the OCSP response.
- RFC 6277 is an update to RFC 2560 addressing the deficiency of the original design which may lead to interoperability failure when the server and the client support different signing algorithms
- Support OCSP client implemented with System SSL (FP0349)

PKI - OCSP

- 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 chooses the signing algorithm to sign the response 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, eg. md-2WithRSAEncryption, md-5WithRSAEncryption, the next one will be used, so on and so forth.
  - 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 - NxM

- 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 - NxM

- 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

PKI - NxM

Example – request for SCEP certificate requires 2 administrator's approval

pkiserv.tmpl

```
<TEMPLATE NAME=5-Year SCEP Certificate - Preregistration>

....
<PREREGISTER>
<ADMINNUM=2>
AuthenticatedClient=AutoApprove
SemiauthenticatedClient=AdminApprove
UnauthenticatedClient=Reject
SubsequentRequest=AutoApprove
RenewalRequest=AutoApprove
</PREREGISTER>
</TEMPLATE>
```

pkitmpl.xml

```
<tns:certreq_template nickname="5YSCEPP">
  <tns:certname>5-Year SCEP Certificate – Preregistration</tns:certname>
…
 <tns:AdminNum>2</tns:AdminNum>

…
</tns:certreq_template>
```

UNIX Search Authority

- To make best use of SUPERUSER.FILESYS.CHANGEPERMS and CHOWN to delegate UNIX security administration, it is necessary to grant READ and SEARCH to all directories or grant a higher-than-desired authority such as AUDITOR or SUPERUSER.FILESYS
  - Provide a more granular mechanism to delegate UNIX security administration, avoiding over-authorization
  - Define a new UNIXPRIV resource to control read/search access to all directories
- Need to prevent the execution of all files in a file system, similar to a 'NOEXEC' mount option. Recommended for directories like /tmp, where any user can write files.
  - Provides a RACF control over file execution, complementary to mounting the file system with 'SETUID NO'
  - Provides straight-forward compliance/audit verification
  - Define RACF profile(s) in the new FSEXEC class the denies file execute access to the specific file system(s)

UNIX Search Authority

- Define a new UNIXPRIV profile SUPERUSER.FILESYS.DIRSRCH
  - READ (or higher) access grants user read and search permission to UNIX directories
  - Generics allowed
- Example:

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

PERMIT SUPERUSER.FILESYS.DIRSRCH CLASS(UNIXPRIV)
     ID(appropriate-groups-and-users) ACCESS(READ)

SETROPTS RACLIST(UNIXPRIV) REFRESH
```

- DIRSRCH authority does NOT grant read, write, or execute permission to ordinary UNIX files.
- DIRSRCH authority does NOT grant write permission to UNIX directories.

UNIX Search Authority

- Define a profile in the new FSEXEC class.
  - Profile name must match the FILESYSTEM name specified on the MOUNT statement.
  - Profile name is case sensitive. Generic names are allowed.
  - Update (or higher) access makes the user eligible for file execution, subject to other access checks.
- Example:

```
RDEFINE FSEXEC /tmp UACC(NONE)
    or
RDEFINE FSEXEC OMVS.ZFS.ADMIN.** UACC(NONE)

PERMIT OMVS.ZFS.ADMIN.** CLASS(FSEXEC) ID(USER019 GROUPADM) ACCESS(UPDATE)

SETROPTS CLASSACT(FSEXEC) RACLIST(FSEXEC)
```

- Superuser or auditor privilege does not override FSEXEC denial of access.
- On denial, ICH408I message includes 'ACCESS ALLOWED (FSEXEC ---)'.
- FSEXEC is supported for ZFS and TFS type file systems.
- FSEXEC does not apply to file systems mounted with the '-s nosecurity' option.

RRSF - Dynamic MAIN Switching

- Switching the MAIN system in a multisystem node is a brutal “11”-step manual process that is not feasible to implement for short-term changes.
  - For example, to accommodate an IPL-window on the MAIN system without suffering an “outage”, as perceived by users.
- This process is essentially replaced by the issuance of 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 prior to V2R2

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:

- Issue

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

  - From **any** system in the MSN

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

- Optionally, update the RACF parameter library

RRSF - Dynamic MAIN Switching

When the Multisystem Node is **not** in a sysplex

- From the old (current) MAIN system issue

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

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 SYSTEM.

- From the new MAIN system, issue:

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

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

- Harden the change in parmlib if you expect reIPLs before switching back the change is intended to be “permanent”.
  - Move the MAIN keyword in the TARGET definitions

RRSF – Unidirectional Connections

- It is impossible to prevent a privileged user on a test system from escalating his privilege on a production system when they are connected using RRSF. The honor system applies.
- Now any RRSF node can define another RRSF node such that inbound requests from that node are to be denied
  - Protect against accidental or malicious damage to your production system
  - Demonstrate to an auditor your compliance with your security policy, regardless of the configuration established on the remote node

RRSF – Unidirectional Connections

- Use a new TARGET command keyword when defining a remote node

TARGET NODE(thatnode) DENYINBOUND

- When the remote node is a multisystem node:

TARGET NODE(thatnode) SYSNAME(*) DENYINBOUND

  - SYSNAME(*) is not required; 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

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**
- **SA23-2284 - z/OS UNIX System Services: Messages and Codes**
- **SA23-2281 - z/OS UNIX System Services Programming: Assembler Callable Services Reference**
- **SA23-6843 - IBM Health Checker for z/OS User's Guide**

- **Hot Topics -** http://publibfp.dhe.ibm.com/epubs/pdf/e0z3n110.pdf