IBM Software Group

Enforce Policy Compliance on RACF

IBM Tivoli zSecure Command Verifier

Guus Bonnes
June 19, 2008

Contents of Presentation

- Product Overview
- Need for the product
- How does it work
- Standard command verification
  - Keywords
  - Parameters
  - Auditing
- Extended command functions
- References

What is the RACF Command Verifier?

- Official name: IBM Tivoli zSecure Command Verifier

- Product that enables an installation to
  - Define a security policy for RACF profiles
  - Enforce the security policy during RACF profile manipulation

- Policy profiles are implemented via RACF profiles
  - Access controls who can execute the command
    (if the policy applies to this situation)
  - Almost all keywords and parameters have a matching policy profile
  - Policy profiles are themselves subject to policy

Why RACF Command Verification?

- RACF command authorization lacks granularity
  - SPECIAL can do anything to any profile
  - Group-SPECIAL can do almost anything to a limited set of profiles
  - Owner can do many things to owned profiles
  - Ordinary users can upset procedures
    Ordinary user is also owner of own resources
  - Segment management has no scoping support

- RACF command authorization only cares about the keywords, and not about the keyword values.
  - Also applies to segments (OMVS: UID=0)

Examples Undesirable Commands

- User with System-SPECIAL
  - Self-permit / Self-connect
- User with System-AUDITOR
  - Can activate/suppress all Audit records
- User with Group-SPECIAL
  - Can give anybody Group-SPECIAL
  - Can connect anybody to the group
- Helpdesk users
  - Can change password of STCs / Batch IDs

Examples Undesirable Commands

- Owners of datasets and general resources
  - Can set WARNING mode
  - Can set UACC
  - Can PERMIT anybody
- Owners of users/group
  - Can delete user/group
  - Can connect anybody to group
- Regular users
  - Can change their name
  - Can change their default group (DFLTGRP)

Examples Undesirable Commands

- If you **can** change the OWNER, you can change it to **anybody**
- If you **can** change the UACC, you can change it to **anything**
- If you **can** change the Access List, you can grant access at **any** level to **anybody**

From Correction to Prevention

- Several responses to policy deviations:

- Report only
  - First step
- Report and correct manually before the auditors come in
  - Common approach
- Report (and correct) automatically
  - As implemented by some commercial products
- Prevent deviations from occurring
  - zSecure Command Verifier
  - RYO

Extended Functions

- Insert proper defaults
- Mandatory values for parameters

- Command Audit Trail
- Extra command keywords when convenient
- Temporary authorizations
- Segment management scoping

Product Overview

- Verify RACF commands against installation policies
- Installation policies are defined via RACF general resource profiles
  - Patented technology to translate policy into profile
- Policy verification is **on top** of RACF authorization

  CV can allow, change or reject a RACF command
  - "Allow" is still subject to RACF command authorization
  - "Change" result must be allowed by RACF
  - "Reject" overrules RACF command authorization

- Special CV profiles can allow temporary increase in authorization
  - (Controlled) Temporary System-SPECIAL

- Installed as IRREVX01, invoked for all commands except
  - BLKUPD, RACLINK, RACDCERT, RVARY, Operator commands

Product Overview …

- Policy profiles are result oriented
  - Describe the result for the target profile
    - C4R.DATASET.UACC.**READ**.<dsname>
  - Don’t care about the actual command used
    - One policy profile to control both
      ALTUSER <userid> PASSWORD
      PASSWORD USER(<userid>)
    - One policy profile to control both
      ALTUSER <userid> UACC(<uacc>)
      CONNECT <userid> GROUP(<group>) UACC(<uacc>)
- Policy profiles allow granular specifications
  - Almost all policy profiles have qualifiers to identify target profile
    - C4R.USER.PASSWORD.<owner>.<userid>
    - Generics can be used as well

Product Overview …

- Policy verification only uses access to the profile
  - Ignores special, operations, trusted, privileged
  - Checks UACC and Access List
    - No profile ➔ No policy ➔ Don’t stop
    - NONE/READ ➔ Not Authorized
    - UPDATE ➔ Authorized

- Use special qualifiers for special policies
  - C4R.DATASET.ACL.=RACUID.<access>.<dsname>
    Putting yourself on the Access List

- Use APPLDATA to assign values
  - C4R.DATASET.=OWNER.<dsname>   APPLDATA(‘=HLQ’)
    The owner should be the same as the High Level Qualifier

Example policies

- Warning mode
  - C4R.<class>.ATTR.WARNING.<profile>
    - READ Reset warning mode
    - UPDATE Set warning mode

- Discrete Profiles
  - C4R.DATASET.TYPE.DISCRETE.<profile>
    - UPDATE Create discrete allowed

- Create more specific resource profiles
  - C4R.DATASET.=UNDERCUT.<dsname>

Example policies …

- Owner of dataset
  - C4R.DATASET.=OWNER.<dsname>
  - C4R.DATASET./OWNER.<dsname>
  - C4R.DATASET.OWNER.=HLQ(n)
  - C4R.DATASET.OWNER.<owner>.<dsname>
  - Optional APPLDATA
    - =HLQ
    - =MYOWNER
    - <userid> or <groupid>

- Prevent any change to certain datasets (e.g. System datasets)
  - C4R.DATASET.=NOCHANGE.<dsname>   APPLDATA('LEVEL=nn')

Example policies …

- Naming conventions for new users/groups
  - C4R.USER.ID.=RACUID(n)
  - C4R.USER.ID.=RACGPID(n)
  - C4R.USER.ID.<userid>

- Managing user/group connections
  - C4R.CONNECT.ID.<group>.<userid>
  - C4R.CONNECT.ID.=USERID(n)

- Naming convention General Resource profiles
  - C4R.TCICSTRN.ID.<profile or member>
    Applies to
    - RDEF for TCICSTRN
    - RALT ADDMEM for GCICSTRN

Example policies …

- Refresh RACLISTed profiles
  - C4R.RACF.<class>.RACLIST

READ	REFRESH
UPDATE	RACLIST/NORACLIST

- Manage Global Access Checking

C4R.GLOBAL.**	For RDEF GLOBAL
C4R.GMBR.**	For RALT GLOBAL ADDMEM
C4R.RACF.GLOBAL.**	For SETROPTS GLOBAL

Only give access to your designated GAC specialist

Auditing the policy

- Two types of policy auditing:

- Standard RACF audit of policy profiles via SMF
  - Uses audit settings of policy profile, like
    AUDIT(ALL(READ))
  - Audit record if policy profile used (allow or deny)

- Command Verifier Command Auditing via SMF
  - Uses three Command Level policy profiles
    C4R.PREAUD.*COMMAND*
    C4R.PSTAUD.*COMMAND*
    C4R.ERRMSG.*COMMAND*

Extended Functions: Mandatory Parameter Values

- Override whatever the user has specified
- Only used when “adding” profiles
- Indicated by use of =FIELDNAME as third qualifier
- Implemented for

•OWNER
  •DFLTGRP
  •SUPGROUP
  •User/Group Attributes

•Password Interval
•UACC
•STDATA user/group

- APPLDATA is used to specify the desired value
  - =RACUID
  - =DFLTGRP
  - =SUPGROUP
  - =OWNER
  - <value>

Extended Functions: Insert Proper Defaults

- Similar to Mandatory Values
- Only used when “adding” profiles
- When RACF requires a value, but user doesn’t provide
- Indicated by use of /FIELDNAME as third qualifier
- Example:
  - C4R.USER./PASSWORD.**   APPLDATA(‘RANDOM’)
  - C4R.USER./OWNER.**   APPLDATA(‘=MYOWNER’)
  - C4R.USER./DFLTGRP.**   APPLDATA(‘=OWNER’)

Extended Functions: Convenient Keywords

- LISTDSD
  - Automatically insert GEN when no discrete profile exists
  - C4R.LISTDSD.TYPE.AUTO.<dsname>

- RDEFINE and ADDSD
  - Automatically insert FROM(<current best profile>)
  - Used the profile currently used for the resource
  - C4R.<class>./FROM.<profile>   APPLDATA(‘=BESTFIT’)
  - Use PERMIT afterwards to update Access List

Extended Functions: Segment Management Scoping

- RACF Command authorization for segments is based on
  - System-SPECIAL (all segments all profiles)
  - FIELD class
    - &RACUID in ACL Allowed for own user profile
    - READ Display
    - UPDATE Add and Change
  - No Group-SPECIAL scoping

- C4R.<class>.<segment>./SCOPE
  - Reduces access to segments to the Group-SPECIAL scope
  - Still requires access to profiles in the FIELD class

Extended Functions: Temporary Authorizations

- Two types of temporary authorizations
  - Unconditional System-SPECIAL
  - Conditional System-SPECIAL
- Based on command
  - C4R.<command>.=SPECIAL
  - Most common for list-type of commands

  - C4R.<command>.=CTLSPEC
  - All keywords must be CV-controlled
  - Most common to allow only a single action, like
    - Self-Connect to emergency group C4R.CONNECT.ID.<group>.=RACUID
    - Permit to single application C4R.DATASET.ACL.=RACUID.UPDATE.HLQ.**

Extended Functions: Command Audit Trail

- Retain information about the last change to a profile
- Kept in profile itself
  - When was TSO segment added?
  - When was user connected to group
  - Who issued PERMIT
  - When was profile set to WARNING
- Displayed via
  - Regular LIST command
  - C4RCATMN command
- Controlled via
  - C4R.<class>.=CMDAUD.<type>.<profile-identification>

Extended Functions: Command Audit Trail

- Example:

```
Segment:  CICS    Added on 05.241/03:19 by C4RTEST
                  Changed on 05.241/03:20 by C4RTEST
          TSO     Changed on 05.241/03:19 by C4RTEST
Attrib:   PASSWRD Removed on 05.238/14:24 by C4RTEST
          INTERV  Changed on 05.241/04:42 by C4RTEST
          RESTR   Added on 05.238/14:24 by C4RTEST
Connect:          BCSC Added on 05.238/14:24 by IBMUSER
GrpAttr:  ADSP    BCSC Removed on 05.238/14:24 by IBMUSER
```

Product History

- First version created in 1995 (Consul/CVO)
  - Used command front-ending to intercept commands
  - Originally required writing assembler exits to implement the policy
  - Some sample exits were provided

- Second version created in 1998
  - Uses RACF Common Command exit (IRREVX01)
  - Still required writing assembler exits to implement the policy
  - Many sample exits were provided

- Third version created in 2002 (Consul/zLock)
  - Policy can be defined via RACF Resource Profile
  - Uses patented technology to translate profile into policy

- First IBM version created in 2007 (zSecure Command Verifier)

References

Documentation on the web:

- http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp
- http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc/c4rbcv19.htm

Experiences Implementing Policy Compliance for RACF using IBM Tivoli zSecure Command Verifier

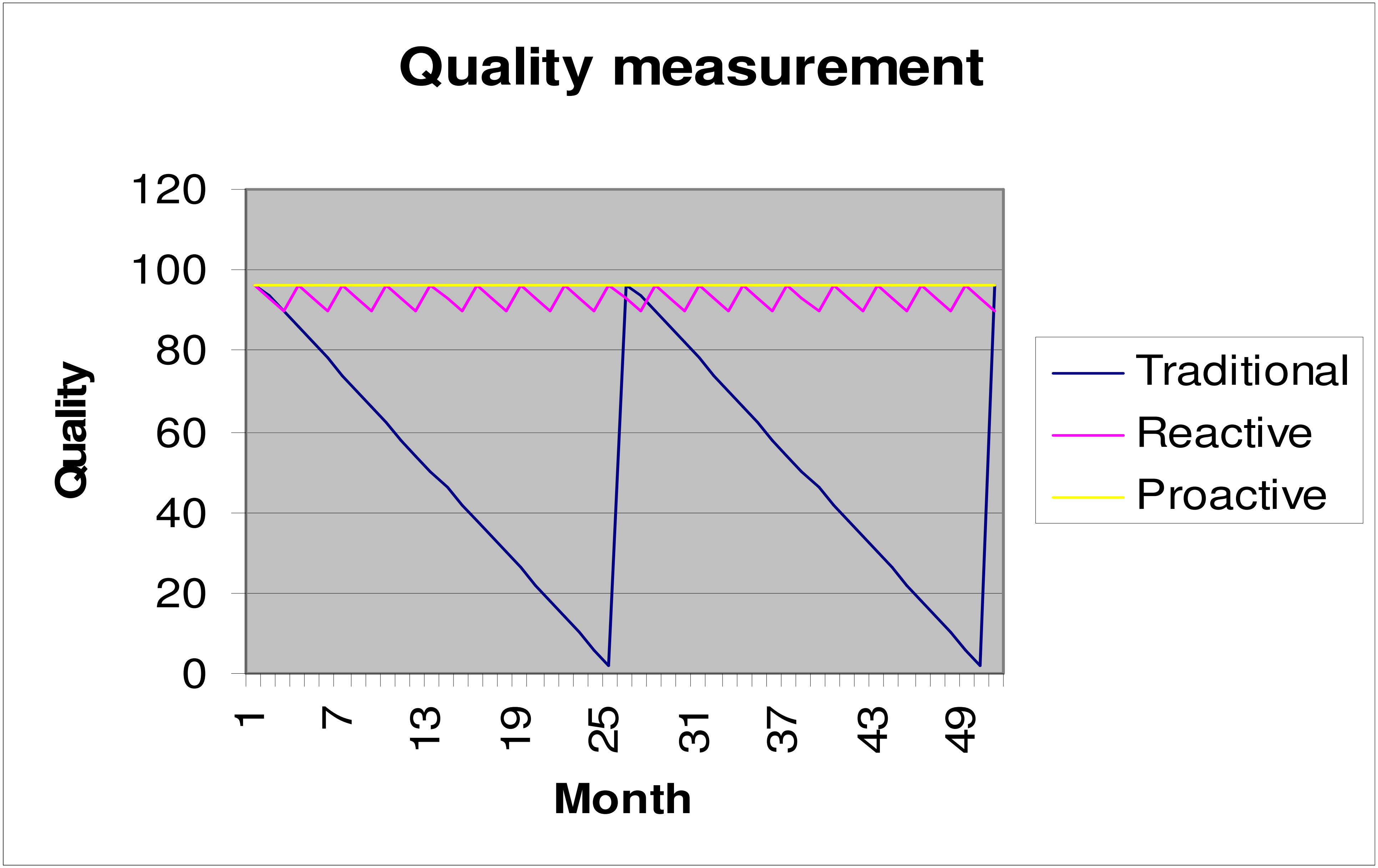
Simon Dodge, CISSP
Senior Consultant
SiCon Inc

SiCon

AGENDA

1. Measuring Quality
2. Challenges & Solutions
3. Audit trail examples
4. Summary

Goal: Minimize deviations in quality



Challenge: Control privileged attributes

- Giving out SPECIAL/OPERATIONS/AUDITOR

- **C4R.USER.ATTR.SPECIAL.*<owner>.<userid>***
- **C4R.USER.ATTR.OPERATIONS.*<owner>.< userid>***
- **C4R.USER.ATTR.AUDITOR.*<owner>.< userid>***

```
alu MAAT special

C4R480E Special attribute not allowed, command
terminated
```

Challenge: Control Group privileges

- Giving out SPECIAL/OPERATIONS/AUDITOR

  - **C4R.CONNECT.ATTR.SPECIAL.<*group*>.<*userid*>**
  - **C4R.CONNECT.ATTR.OPERATIONS.<*group*>.<*userid*>**
  - **C4R.CONNECT.ATTR.AUDITOR.<*group*>.<*userid*>**

```
connect AMUN group(webinar) special

C4R551E GrpSpecial attribute not allowed, command
terminated
```

Challenge: Controlling Generic / Discrete profiles

- Desired no DISCRETE profiles in DATASET class

- Desired no GENERIC profiles in some classes

- **C4R.<class>.TYPE.DISCRETE.<*profile*>**
- **C4R.<class>.TYPE.GENERIC.<*profile*>**
  - UPDATE would allow you to create profile

```
addsd 'ANUBIS.discrete'

C4R613E DISCRETE profiles not allowed, command
terminated
```

Challenge: Abuse of WARNING mode

- Desire to restrict who can turn WARNING on

- **C4R.<*class*>.ATTR.WARNING.<*profile*>**
  - **UPDATE allows you to set warning**

```
altdsd ‘OSIRIS.**' warning

C4R611E Warning mode not allowed, command terminated
```

Challenge: Prevent Abuse of UAUDIT

- Many non administrators had AUDITOR
  - For problem diagnostics
  - Concern that they could change UAUDIT setting

- **C4R.USER.ATTR.UAUDIT.<*owner*>.<*userid*>**
  - Checked for both setting & removing UAUDIT

```
alu HATHOR uaudit

C4R486E Uaudit attribute not allowed, command
terminated
```

Challenge: Pre EGN format profiles

- Activated EGN 4 years ago; Many folks still create new dataset profiles `hlq.*.**` ☹

- Control creation of `hlq.*.**`
  - `C4R.DATASET.**.+.++`

```
ADDSD `ISIS.TMP.*.**'

C4R640E Define/Delete DATASET ISIS.TMP.*.** not
allowed, command terminated
```

Challenge: Excessive public access / permits

- Excessive use of UACC > READ
  - Easier than determining who needs access

- **C4R.<*class*>.UACC.<*uacclevel*>.<*profile*>**
  - **EG C4R.DATASET.UACC.ALTER.****

```
altdsd `SETH.**' uacc(alter)

C4R600E UACC ALTER setting not allowed, command
terminated
```

Challenge: Installation data ‘*corruption*’

- Installation data on userids intended to have specific information; Was being modified inappropriately by decentral administrators

- Implemented a check against a RACF profile like
  - **C4R.USER.INSTDATA.*<owner>*.*<userid>***

```
alu THOTH data('Can I update installation data ?')

C4R513E Instdata change not allowed, command
terminated
```

Challenge: Use of certain Ids in Access lists

- Prevent Group STCCA7 being used in a PERMIT command

- Permits for **`ID(STCCA7)`** controlled via:
  - **`C4R.*.ACL.STCCA7.**`**

- Tremendous granularity:
  - **C4R.<*class*>.ACL.<*id*>.<*access*>.<*profile*>**

```
permit ‘RA.**' id(stcca7) access(read)

C4R601E ACL setting STCCA7 READ not allowed, command
terminated
```

Challenge: Production Support team

- Dedicated team for production profiles
  - All PROD administration goes via their team
  - Can force consistency via a single team

- Profile creation controlled via:
  - `C4R.DATASET.ID.PROD%%%.**`
- Profile UACC controlled via:
  - `C4R.DATASET.UACC.PROD%%%.**`
- Permits controlled via:
  - `C4R.DATASET.ACL.*.*.PROD%%%.**`
- Warning controlled via:
  - `C4R.DATASET.ATTR.WARNING.PROD%%%.**`

Challenge: Restricting who can manage policies

- In this example, policy profiles are stored in **`$POLICY`** class
  - Must restrict RACF policy changes to “Engineering team”

- Control who can manage policies via:
  - **`C4R.$POLICY.**`**

- Control who can issue SETROPTS for class via:
  - **`C4R.RACF.$POLICY.**`**

- Control activation/inactivation of exit via:
  - **`FACILITY  CSVDYNEX.IRREVX01.*`**

Challenge: “Some” special but not all

- One team is allowed to change <u>anyone’s</u> OWNER
  - <u>Without</u> having SPECIAL

- UPDATE to **`C4R.<command>.=CTLSPEC`** will grant you special ,
  - ***Only*** for the duration of the specific ***`<command>`*,**
  - ***And if*** all keywords are controlled by CV

- **`C4R.ALTUSER.=CTLSPEC`**
- **`C4R.USER.OWNER.**`**

Sample audit trail - Userid

```
USER=ANUBIS  NAME=GUESS WHO             OWNER=SECADMIN    CREATED=03.232
… Lines snipped …
SECURITY-LABEL=NONE SPECIFIED
C4R736I Command Audit Trail for USER ANUBIS
C4R739I Segment:  CICS    Added on 06.087/16:28 by SEKHMET
C4R739I           OMVS    Added on 08.053/10:10 by ODIN
C4R739I           WORK    Added on 06.087/16:29 by SEKHMET
C4R739I Attrib:   UAUDIT  Removed on 07.332/15:06 by SEKHMET
C4R739I                   Added on 07.332/15:21 by GEB
C4R739I           AUDITOR Removed on 07.303/10:33 by SEKHMET
C4R739I                   Added on 07.313/11:37 by GEB
C4R739I           PASSWRD Added on 06.283/15:53 by ISIS
C4R739I           RESUME  Added on 06.283/15:54 by ISIS
C4R739I           OWNER   Changed on 08.108/09:16 by OSIRIS
C4R739I           DFLTGRP Changed on 08.108/09:16 by OSIRIS
C4R739I           NAME    Changed on 08.120/11:19 by NUT
C4R739I Connect:          RC1772 Removed on 07.190/12:39 by PROMETHU
C4R739I                   SYS1 Removed on 07.213/12:43 by NUT
C4R739I                   @SECLSE Added on 07.298/12:34 by NUT
C4R739I                   EMPL Removed on 07.298/17:26 by NUT
C4R739I                   @TSD Removed on 07.303/10:35 by ANUBIS
C4R739I                   $U21AS Added on 08.108/09:16 by OSIRIS
C4R739I GrpAttr:  SPEC    @TSD Removed on 07.303/10:31 by ANUBIS
C4R739I                   @SECLSE Removed on 07.303/11:22 by ANUBIS
C4R739I           OPER    @TSD Removed on 07.303/10:31 by ANUBIS
```

Sample audit trail – Dataset profile

```
LISTDSD DA('HERA.**')
… Lines snipped …
NO ENTRIES IN CONDITIONAL ACCESS LIST

C4R736I Command Audit Trail for DATASET  HERA.**
C4R739I Attrib:   WARNING Added on 08.072/11:07 by ZEUS
C4R739I                   Removed on 08.072/11:07 by ZEUS
C4R739I Access:           @SECLSE access READ on 07.347/10:11 by AMANRA

C4R739I                   FRED access READ on 08.093/08:56 by ISIS
```

Summary

- You can now control things that you just couldn’t even conceive of before
- Management of policies is via familiar RACF profiles
  - No external configuration to define a policy
- You can now sleep better at night
- I very much recommend supplementing with SMF based reports to show activity/violations

Thank You for Joining Us today!

Go to **www.ibm.com/software/systemz** to:

- Replay this teleconference
- Replay previously broadcast teleconferences
- Register for upcoming events