z/OS 2.4 IBM Education Assistant (IEA)

Solution (Epic) Name: RACF ACEE privilege escalation detection

Element(s)/Component(s): RACF/SAF

Agenda

- Trademarks
- Session Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Session Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Session Objectives

- Describe the privilege escalation detection function
  - Overview
  - Activation
  - New message
  - Exception processing
  - Command exit enhancement
  - Failure option
  - Recap

Overview

- Who (Audience)
  - Security administrators, security auditors
- What (Solution)
  - Function to detect authority-related modifications to the ACcessor Environment Element (ACEE) since the time it was created by SAF or RACF
  - New IRR421I message issued to security console containing environmental information at time of detection
- Wow (Benefit / Value, Need Addressed)
  - Useful in detecting programs that fall outside your security policy
  - Useful in detecting programs that might be requesting more privilege than absolutely necessary

Overview …

- What is an ACEE?
  - The ACcessor Environment Element (ACEE) is the user’s ‘security credentials’
  - A piece of storage created when the user logs on
  - The contents are derived from information in the USER profile, containing the user ID, list of groups, various authorities (SPECIAL, AUDITOR, etc), and lots of other stuff describing the user
  - It is used in RACF commands and authorization checking to determine authority/access
  - It is anchored on the address space, or task, or created by an authorized application and passed explicitly to various SAF services

Overview …

- What type of application would modify an ACEE?
    - A perfectly well-behaved and well-intentioned one that has no good alternatives
    - A perfectly well-behaved and well-intentioned one that could, in fact, better use features of RACF to make the modification unnecessary
    - A perfectly well-intentioned one that nonetheless does not adhere to the principle of least privilege
    - A customer-written program that may or may not fall within the security policy of that installation (e.g. a system programmer's "productivity aid" in the form of a "magic SVC")
    - Malware planted by an insider or intruder
    - Malware exploiting a vulnerability in system software to regain control in an authorized state

Overview …

- A ‘fingerprint’ is created when an ACEE is created with RACROUTE REQUEST=VERIFY/X
- The fingerprint encapsulates the user ID and various authority-related fields, such as SPECIAL, OPERATIONS, PRIVILEGED, TRUSTED
- When the new ACEECHK class is active
  - The fingerprint is verified at the beginning of RACROUTE REQUEST=AUTH and most RACF commands
  - New message IRR421I is issued when privilege escalation is detected
    - The function continues to completion as normal
  - IRR421I can be suppressed for trusted programs by defining exceptions (profiles) in the ACEECHK class

Usage & Invocation

- SETROPTS CLASSACT(ACEECHK) RACLIST(ACEECHK)
- User TSOUSR8, with a modified ACEE, issues:

```
ADDUSER GAIL
```

- On the console:

```
IRR421I ACEE modification detected
 for user TSOUSR8 in address space ID 0x00000026 running under user
 TSOUSR8 and job name TSOUSR8 while program ADDUSER  is running.  The
 RACF function detecting the modification is IRRENV00.
 Rsn=0x60000000.  (ACEEPRIV is ON) (ACEESPEC is ON).  Occurrences 1.
 Command=ADDUSER.  Call chain: ADDUSER <- IKJEFT02 <- IKJEFT01
```

- The call chain contains every program under the current TCB, plus the first program of all ancestor TCBs, up to and including the job step TCB.

Usage & Invocation …

- **And if I bless this particular modification?**
  - Create an exception in the ACEECHK class for the program making the modification
  - For example, TESTPROG modifies the ACEE, eventually calls OTHERPGM, which accesses some resource that drives a SAF check

    ```
    RDEFINE ACEECHK IRR.EXCLUDE.TESTPROG
    SETROPTS RACLIST(ACEECHK) REFRESH
    ```
  - RACF will check "up the call chain" (as described on the previous slide) to the job step TCB for an exception

  - This is an existence check; the access list, UACC, etc. is irrelevant

Usage & Invocation …

- Exception list – additional details
  - Program name can be further qualified by one or two user IDs for more granularity
    - Example 1: A batch job should only be granted the exception when run by the user ID intended to be used for that job

```
RDEFINE ACEECHK IRR.EXCLUDE.BATCHPGM.BATCHUSR
```

    - Example 2: A server program attaches tasks to run under end-user authority. The exception should only be granted when the server is running as a specific user ID, performing for a specific client.

```
RDEFINE ACEECHK IRR.EXCLUDE.SERVERPG.SERVER01.USER01
```

  - If program control is active, the list is only checked if the environment is clean
  - If the user ID was changed, any exception profile containing that user ID is ignored

Usage & Invocation …

- The integrity of the ACEECHK class is defended against a modified ACEE by failing:
  - Attempts to add an ACEECHK exception using the RDEFINE command.
    - IRR421I issued to console
    - User sees **ICH10103I NOT AUTHORIZED TO DEFINE IRR.EXCLUDE.MYPROG.**

  - ***Any*** SETROPTS command
    - IRR421I issued to console
    - User sees **ICH14001I NOT AUTHORIZED TO ISSUE SETROPTS.**

Usage & Invocation …

- Actual example with IBM DB2 Log Analyzer

```
IRR421I ACEE modification detected
 for user XCB4025 in address space ID 0x000001B5 running under user
 XCB4025 and job name YYIBMLOG while program ALAGEN1 is running.  The
 RACF function detecting the modification is IRRRCK00.
 Rsn=0x40000000.  (ACEEPRIV is ON).  Occurrences 1.
 Resource=GNO10NJE.XCB4025.YYIBMLOG.JOB06667.D0000120.?(JESSPOOL).
 Call chain: ALAGEN1
```

- This was a simple batch job with only one program in the chain.
- ACEEPRIV is turned on so that log files are accessed without an audit trail

Usage & Invocation …

- Actual example with zSecure Collect

```
IRR421I ACEE modification detected
 for user BCSCGB1 in address space ID 0x0000001B running under user
 BCSCGB1 and job name CKFCOLL1 while program CKFCOLL is running.  The
 RACF function detecting the modification is IRRRCK00.
 Rsn=0x40000000.  (ACEEPRIV is ON).  Occurrences 1.
 Resource=IFASMF.DEFAULT(LOGSTRM ).  Call chain: CKFCOLL
```

- This program has a valid reason to read many z/OS resources. It does not run as a started task, and so cannot obtain the PRIVILEGED attribute without modifying the ACEE. An exception is recommended.
  - RDEFINE ACEECHK IRR.EXCLUDE.CKFCOLL

Usage & Invocation …

- Another example with zSecure Collect

```
IRR421I ACEE modification detected
 for user IBMUSER in address space ID 0x0000001B running under user
 BCSCGB1 and job name CKFCOLL1 while program DSN3ID00  is running.
 The RACF function detecting the modification is IRRRCK00.
 Rsn=0x40008000.  (ACEEPRIV is ON) (ACEEUSRI: expected BCSCGB1, actual
 IBMUSER ).  Occurrences 1.  Resource=DBCG.BATCH(DSNR    ).  Call
 chain: DSN3ID00 <- DSNUTILB <- CKFCOLL
```

- In this example, the program that modified the ACEE (CKFCOLL) was not the program running (DSN3ID00) when the SAF check detected the modification.
  - DSN3ID00 is a DB2 program accessing DB2 resources. It would not be appropriate to create an exception for the DB2 program(s), but we can see that they were called by CKFCOLL.

Usage & Invocation …

- Actual example with DFSMShsm

```
IRR421I ACEE modification detected
 for user IBMUSER in address space ID 0x00000037 running under user
 IBMUSER and job name -None- while program ARCCTL   is running.
 The RACF function detecting the modification is IRRRCK00.
 Rsn=0x40000000. (ACEEPRIV is ON) Occurrences 1. Resource=DFHSM(TAPEVOL).
 Call chain: ARCCTL
```

- This program has a valid reason to bypass security checks. But it runs as a started task, and should not need to modify the ACEE, with the proper STARTED task profile/user defined.
- hsm shipped OA54740 to beef up their documentation, and to provide an option to run with TRUSTED instead of PRIVILEGED.
  - If the proper setup has not been performed, they will continue to modify the ACEE to maintain the current behavior, and IRR421I messages will result.

Usage & Invocation …

- A contrived example in a dirty environment

```
IRR421I ACEE modification detected
 for user TSOUSR8 in address space ID 0x00000026 running under user
 TSOUSR8 and job name TSOUSR8 while program LU  is running.  The RACF
 function detecting the modification is IRRENV00.  Rsn=0x60002000.
 (ACEEPRIV is ON) (ACEESPEC is ON).  Occurrences 1.  Command=LU.
 Profiles in the ACEECHK class were ignored because the execution
 environment is not clean. Call chain: LU <- IKJEFT02 <- IKJEFT01
ICH420I PROGRAM IRXANCHR FROM LIBRARY SYS1.LINKLIB CAUSED THE
 ENVIRONMENT TO BECOME UNCONTROLLED.
ICH420I PROGRAM LU FROM LIBRARY RACFDRVR.BRWELLS.R24.LINKLIB CAUSED
 THE ENVIRONMENT TO BECOME UNCONTROLLED.
```

Usage & Invocation …

- RACF’s **Common Command Exit (IRREVX01)** can be used to temporarily escalate privilege (e.g. by modifying the ACEE in the pre-exit so the command runs with AUDITOR).
- This could trigger IRR421I
- So, the exit is enhanced with new options to request that the command run with SPECIAL or AUDITOR attributes, without the exit having to modify the ACEE itself

Usage & Invocation …

- A new output parameter field is added to the exit's parameter list
- Pre-exit can request that the command run with SPECIAL and/or AUDITOR authority
- RACF will
  - Modify the ACEE
  - Re-calculate the fingerprint, in case the command drives any authorization checks, or post-exit runs a 'nested command'
  - Run the command
  - Call the post-exit
  - Remove the temporary authority
  - Restore the original fingerprint
- The post-exit has no responsibility to remove the authority

Usage & Invocation …

- Interface details
  - Exit p-list mapped by IRREVXP mapping macro, documented in RACF Data Areas as EVXP
  - P-list is a list of addresses which point to data
  - The first address points to the number of fullwords (addresses) in the p-list (including itself)
  - If this value is >= 12, the new function is available
  - The 12th address (offset 44 in EVXP) points to a fullword bitstring (EVXOPARM) initialized to 0s. Pre-exit can set bit 0/1 as follows:

X’80000000’	**EVXSPEC**	**Pre-exit requests to run the command with system SPECIAL authority**
X’40000000’	**EVXAUDT**	**Pre-exit requests to run the command with system AUDITOR authority**

Usage & Invocation …

- Failure option
  - If, after running with ACEECHK active, and feeling very confident that you have identified all known ACEE modifiers, and if you feel very lucky …
  - If you define the **IRR.ABEND.ON.FAILURE** profile in the ACEECHK class, and no exception profile exists or can be used, then abend 4C6 (existing) is issued with (new) reason code 2766 (X'ACE')

  - The extent of the fallout will depend on the recovery routines in effect for the programs that are running

Interactions & Dependencies

- To exploit this item, all systems in the Plex must be at the new z/OS level: No
- Software Dependencies
  - DFSMShsm – consider OA54740, which allows option to run TRUSTED instead of privileged and avoid IRR421I
- Hardware Dependencies
  - None
- Exploiters
  - zSecure Command Verifier will exploit the command exit enhancement

Migration & Coexistence Considerations

- Look for conditioning APAR OA56850 (z/OS V2R2 and V2R3)
- Look for rollback APAR OA56851 (z/OS V2R3 only)

Installation

- List anything that a customer needs to be aware of during installation and include **examples** where appropriate - clients appreciate these:
  - No considerations

Session Summary

- Activating the ACEECHK will identify programs that modify ACEEs
- Such programs are not necessarily misbehaved
- Detection is not guaranteed to detect a malicious program running in an authorized state
- Exceptions can be added to suppress IRR421I for trusted programs
- The RACF Common Command Exit can request that a given command run with SPECIAL or AUDITOR authority, and this will not trigger IRR421I
- A failure mode can be enabled to abend programs when a modification is detected

Appendix

- Bruce Wells (brwells@us.ibm.com)

- **Security Server RACF Security Administrator's Guide**
- **Security Server RACF Messages and Codes**
- **Security Server RACF System Programmer's Guide**