zSeries Security Update 2005

z/OS V1R7 Security Server

RACF Enhancements

Redbooks

International Technical Support Organization

zSeries Security Update 2005

z/OS V1R7 Security Server

RACF User Extensions Enhancements

Redbooks

International Technical Support Organization

Session Objectives

- Understand why this new support is provided
- Description of updates
- Migration concerns
- Usage information
- Validation request
- Helpful references

Problem Summary

- There are a great number of outstanding requirements regarding the processing of user and group profile information.
- This new support implements solutions to many requested functional updates to user and group processing concern.
- Long standing RACF requirements have been answered and all installations can take advantage of these updates.

Answered Requirements

- Common Criteria requires a larger number of possible passwords. It lesser the chance of passwords being compromised by brutal force attack.
  - Probability of guessing password less than 1 in 2.5 times 10 to the 14th
- SMF logging of password changes. All password changes cause logging, allowing a tighter auditing.
- Minimum time before next password change is allowed. It provides more secure passwords by enforcing history.
- Revoke Date field not displayed after it is surpassed.

Answered Requirements …

- Keep user ID Revoke Date after RESUME command. It allows for better understanding of why a user was revoked.
  - Do not automatically reset REVOKE and RESUME dates

- Use user ID create date during INACTIVE processing. INACTIVE processing extended to cover new users, allowing for revocation of unused new IDs.

- Display create date when group is listed (LISTGRP). It assists with group management and is consistent with other profile types.

Mixed Case Password Support

- Extend the password rules for lower case characters
- Have switch to enable mixed case passwords
- Allow lower case characters in commands and macros for passwords

Extend Password Rules

- Character types extended on SETROPTS
  - ALPHA – Upper case letters and # $ @
  - ALPHANUM – ALPHA and 0-9
  - VOWEL – Upper case AEIOU
  - NOVOWEL – ALPHANUM – VOWEL
  - CONSONANT – ALPHA – VOWEL
  - NUMERIC – 0-9
  - NATIONAL - # $ and @
  - MIXEDCONSONANT – Upper and lower case CONSONANT
  - MIXEDVOWEL – Upper and lower case VOWEL
  - MIXEDNUM – Upper and lower case ALPHANUM

RULE1(LENGTH(6) NATIONAL(3) ALPHA(4:6))

Mixed Case Password Support

- Have switch to enable mixed case passwords
  - Activated by keyword on SETROPTS
  - PASSWORD(MIXEDCASE|NOMIXEDCASE)
  - Once set to MIXEDCASE should not be reset
- Allow lower case characters in commands and macros for passwords
  - ADDUSER, ALTUSER, PASSWORD, RACLINK, RACROUTE (VERIFY, VERIFYX)
  - If MIXEDCASE set will not fold passwords to upper case
  - If a password has never been set since turning on MIXEDCASE, an extra check will be done for an uppercase password during verification checking

SMF Logging Support

- SMF logging of password changes
  - SETR AUDIT(USER)
    - Will log successful password changes

Minimum Password Change Interval Support

- Disallow password change before interval specified
- Set by SETROPTS PASSWORD(MINCHANGE(interval))
  - Interval is days before password change allowed (0-254)
  - Overridden by SPECIAL or CONTROL access to IRR.PASSWORD.RESET
  - ALTUSER, PASSWORD, RACROUTE (VERIFY, VERIFYX)

Revoke/Resume Support

- Revoke and Resume Date processing updated
  - When a user or connection is RESUMEd or REVOKEd RESUME and REVOKE dates will remain in profile
  - Dates cleared only when
    - NOREVOKE|NORESUME on ALTUSER or CONNECT command
    - REVOKE specified and both REVOKE and RESUME dates are in the past and RESUME date is after REVOKE date

Creation Date Processing Updates

- Inactivity checking to start at user creation
  - ADDUSER now stores creation date in LJDATE field which is used for inactivity checking
- Creation date displayed by LISTGRP

Miscellaneous Updates

- R_admin
  - Updated to mimic command updates
- RCVT
  - RCVTPLC – Bit to allow lower case passwords
  - RCVTPMIN – Byte containing min password change interval
- IRRTEMP2 (database templates)
  - Add definition of PASSASIS field in User profile

PASSASIS 086 20 80 00000001 00 Mixed case pwd

Migration/Coexistence Considerations

- Before issuing SETROPTS(MIXEDCASE) applications supplying passwords must not automatically uppercase passwords

- When sharing a database, all systems processing users with mixed case passwords should be at a support level that supports SETROPTS(MIXEDCASE)

Usage & Invocation

- Commands updated
  - SETROPTS, ADDUSER, ALTUSER, LISTGRP
- Callable interfaces updated:
  - RACROUTE, R_admin
- New Messages
  - SETROPTS
    - ICH14083I Minimum change interval exceeds the password interval
  - ALTUSER
    - ICH21036I Password change rejected due to installation password change interval
  - PASSWORD
    - ICH08017I Password change rejected due to installation password change interval

Session Summary

- What we have covered:
  - Why we provided this new support
  - Description of updates
  - Migration concerns
  - Usage information

Updated publications

- *z/OS Security Server RACF Command Language Reference - SA22-7687*
- *z/OS Security Server RACF Messages and Codes - SA22-7686*
- *z/OS Security Server RACROUTE Macro Reference - SA22-7692*
- *z/OS Security Server RACF Macros and Interfaces - SA22-7682*
- *z/OS Security Server RACF Security Administrator's Guide - SA22-7683*
- *z/OS Security Server RACF Auditor's Guide - SA22-7684*
- *z/OS Security Server RACF Callable Services - SA22-7691*
- *z/OS Security Server RACF Data Areas - GA22-7680*
- *z/OS Security Server RACF Diagnosis Guide - GA22-7689*

zSeries Security Update 2005

z/OS V1R7 Security Server

RACF PassTicket Extensions

Redbooks

International Technical Support Organization

Session Objectives

- Brief introduction to PassTicket.
- Describe new interfaces to RACF PassTicket technology.

Introduction to RACF PassTicket

- The RACF PassTicket is a one-time-only password that is generated by a requesting product or function. It is an alternative to the RACF password that removes the need to send RACF passwords across the network in clear text.
- A PassTicket is only valid for a few minutes before it expires.
- PassTicket functionality was added to RACF about 10 years ago.

Overview of new support

- Previously, the use of RACF to generate PassTickets was restricted to 31 bit supervisor state callers which called a legacy branch-entered assembly code routine.
- In z/OS V1R7, new callable service interfaces are provided to allow non-supervisor state callers to access PassTicket generation and evaluation function if they pass certain authorization checks. Requests from both 31 and 64 bit callers are acceptable.
- Java classes are also provided giving Java caller the ability to generate or evaluate RACF PassTickets.

PassTicket Review

- In order to use PassTickets, one or more profiles are created in the PTKTDATA class using the RDEFINE command. A secret key is specified in the SSIGNON segment in this profile.
- Each profile corresponds to a specific application for the PassTicket. These applications include APPC, CICS, IMS, TSO, MVS batch and VM, as well as custom applications.
- An application generates a PassTicket using either a RACF service or implements the published PassTicket algorithm as documented in "**z/OS Security Server RACF Macros and Interfaces**".
- Application then logs into z/OS (RACF) supplying userid and PassTicket & application and the PassTicket is evaluated.

PassTicket Review

- The PTKTDATA profile contains a secret key. The secret key must be shared between the application which generates the PassTicket, and RACF. The PassTicket generation algorithm is documented in "**z/OS Security Server RACF Macros and Interfaces**".
- PassTicket functionality can be disabled by deleting all PTKTDATA profiles.
- Supervisor state, key 0 callers have always been able to call a 'legacy' branch entered function to generate PassTickets.

New PassTicket Extensions

- RACF callable services in RACF allow problem state callers (who have been granted access) to generate and evaluate PassTickets.
- Java interfaces to allow Java applications to generate and evaluate PassTickets.
- PassTicket Generation and Evaluation is now audited.

What has not changed

- The legacy branch-entered service continues to function as before. No changes need to be made to applications which use this interface.

- Auditing has been enabled for applications which call the legacy branch-entered service.

** Installations which use the legacy service should verify the auditing options on PTKTDATA profiles. The auditing options were previously unused.

Callable Service details

- Two existing callable services were extended to perform PassTicket operations.
- R-Ticketserv generates and evaluates PassTickets for 31 bit applications.
- R_GenSec generates and evaluates PassTickets for 64 bit applications.
- Both services contain the same functionality. The only differences are in the AMODE of the caller and parameter lists of the services.

Java

- Java applications may now use the new IRRPassTicket class to generate and evaluate RACF PassTickets.
- The IRRPassTicket class is found in /usr/include/java_classes/IRRRacf.jar.
- IRRPassTicket uses native methods (JNI) to call r_tickerserv and/or r_gensec to perform PassTicket operations.
- JavaDoc documentation for the IRRPassTicket is located in /usr/include/java_classes/IRRRacfDoc.jar, which must be copied to a workstation.

** Use FTP to retrieve /usr/include/java_classes/IRRRacfDoc.jar to a workstation. Then use the java 'jar' utility to decompress the file. The resulting HTML documentation is readable with most web browsers.

PassTicket evaluation

- The PassTicket evaluation service only evaluates that a PassTicket is computationally valid for a given userid and application. It does not actually log the user in to the system or create any kind of z/OS security context for that user.
- To log in a user using a PassTicket, use a standard z/OS function such as __login() or RACROUTE REQUEST=VERIFY.
- The intent is to allow temporary userids and PassTickets to be generated for userids which do not exist in RACF, but can still be evaluated to determine that they come from a trusted source. That is, a source which shares the secret PassTicket key.

Auditing

- Successful PassTicket generation requests and all PassTicket evaluations can be audited by setting the audit options on PTKTDATA profiles, or by setting LOGOPTIONS for the PTKTDATA class.
- The SMF records will contain the userid of the requestor of the PassTicket operation, the application, and the userid for whom the PassTicket operation was requested.
- Audit records for successful PassTicket generation are now also created when the legacy branch-entered PassTicket generation service is used.
- No new audit records will be created when a user logs in using a PassTicket.

Authorization for Java and Callable service users

- New profiles in the PTKTDATA class control who is permitted to use r_ticketserv and r_gensec to perform PassTicket operations.
- These profiles are in the form IRRPTAUTH.*application.target_userid*
- T*arget_userid*, is a user and may not be a group.
- The callable service caller who wishes to perform a PassTicket generation for *target_user* on *application* must have UPDATE access to the profile.
- The callable service caller who wishes to evaluate a PassTicket generated for *target_user* on *application* must have READ access to the profile.
- Generic profile are permitted, for example IRRPTAUTH.**, or IRRPTAUTH.APPL1.**.
- All callers, regardless of state (supervisor or problem) or key are subject to the authorization check.
- The authorization check is not performed in the legacy branch-entered service.

Authorization for Java and Callable service…

- Granting UPDATE permission to an IRRPTAUTH.*application.target-userid* gives a user the ability to generate a PassTicket for another user. In effect, they can access to the system using another users identity. Careful consideration must be made when granting this access.
- The PassTicket evaluation function is meant to be used to evaluate PassTickets for userids which do not exist RACF. For example, temporary or generated userids, although the evaluation function will work for userids which do exist in RACF.
- There is no revocation of userids due to failed PassTicket evaluation attempts when using the callable services, so care must be taken in granting READ access to IRRPTAUTH.*application.target-user* profiles in the PTKTDATA class. This is to prevent a malicious user from trying to guess a PassTicket by trying repeatedly.

Session Summary

- New callable service interfaces to existing PassTicket generation.
- PassTicket evaluation functionality for use by applications.
- New services are useable by non supervisor state/key 0 applications
- Java interfaces to PassTicket functions.
- Auditing added to legacy and new PassTicket functions.

Publications

Additional information about RACF PassTickets can be found in

- *z/OS Security Server RACF Macros and Interfaces - SA22-7682*
- *z/OS Security Server RACF Security Administrator's Guide - SA22-7683*

Detailed documentation for r_ticketserv and r_gensec callable services can be found in

- *z/OS Security Server RACF Callable Services - SA22-7691*

zSeries Security Update 2005

z/OS V1R7 Security Server

RACF R_Admin Extract

Redbooks

International Technical Support Organization

Session Objectives

- Describe new extract functions of the RACF R_admin callable service.

Overview

- Using R_admin to retrieve profile information was inefficient.
- New extract function codes have been added to allow programs to retrieve User, Group and Connect information from the RACF database.

R_admin overview

- R_admin is a RACF callable service, originally created to give applications api-like access to add, alter, delete and list commands.
  - ADDUSER, ALTUSER, DELUSER, LISTUSER
  - ADDGROUP, ALTGROUP, DELGROUP, LISTGROUP
  - RDEFINE, RALTER, RDELETE, RLIST
  - ADDSD, ALTDSD, DELSD, LISTDSD
  - CONNECT, REMOVE, PERMIT
  - SETROPTS
- Applications create a parameter list, and call R_admin.
- R_admin builds a RACF command image and sends it to the RACF Address Space (RASP).
- Output from the command is returned to the caller.

Problems with ‘list’ functionality of R_admin

- In order to retrieve information from RACF, applications must to parse command output from list commands such as LISTUSER.
- Command output is limited to a max of 4096 lines of output. It is possible for the output of LISTUSER and LISTGROUP (in particular) to be longer, causing loss of data.
- All commands are processed by the RASP. The RASP can be overload by too many R_admin requests and run out of certain limited memory resources.
- Transferring all requests to the RASP for execution, then getting output back is inefficient.

Getting information through an application

- The underlying problem is there is no good way for an application to get profile information from the RACF database.
- The output from ‘list’ commands is not a programming interface. Even if it was, parsing text is inefficient and error prone.
- The other option is to use RACROUTE or ICHEINTY to query the RACF database. This requires too much detailed information about RACF internals. Again, much harder than it needs to be.

What is new in R_admin?

- New function added to R_admin callable service to allow callers to extract User, Group and Connect information from the RACF database in tokenized form, easily readable by software.
- The new r_admin extract function runs in the callers address space, and is not sent to the RASP for processing. This improves performance, and eliminates the problem of overloading the RASP.
- All data in the profile is returned. There is no limit to the amount of data returned.
- R_admin documentation has been improved.
- The ‘list’ functionality is unchanged. Applications which use ‘list’ functionality will continue to function.

What is unchanged in r_admin

- R_admin update (Add, alter, delete, list) requests are still sent to the RASP as before.
- The new extract function is limited to User, Group and Connect information. R_admin does not extract Dataset or General Resource information
- Digital certificate information and RACLINK associations are not extracted by R_admin.

RACROUTE EXTRACT & LIST commands

- R_admin extract can be viewed as a combination of a RACROUTE REQUEST=EXTRACT and a ‘list’ command.
- Data is extracted in a machine readable format like RACROUTE REQUEST=EXTRACT.
- Authorization to extract a given profile is dictated by the same rules as the ‘list’ commands. Problem state callers may call R_admin extract and get back the same data that a ‘list’ command would allow.
- Most returned data is formatted such that it can be specified on an add/alter command.
  - For example, dates are returned in the same MM/DD/YY format that is required on a RACF ALTUSER/ALTGROUP command.

Usage & Invocation

- R_admin extract can be invoked by calling existing RACF callable service IRRSEQ00 with one of the new function codes.

  - **25 ADMN_XTR_USER - Extract specified user.**
  - **26 ADMN_XTR_NEXT_USER - Extract user 'after' specified user.**
  - **27 ADMN_XTR_GROUP - Extract specified group.**
  - **28 ADMN_XTR_NEXT_GROUP – Extract group 'after' specified group.**
  - **29 ADMN_XTR_CONNECT – Extract connection information between specified user and group.**

- The ADMN_xxx_NEXT functions can be used to iterate through all users or groups in the database by calling R_ADMIN in a loop, passing in the previously retrieved used/group id. Pass in a BLANK id on the first call.

What data is returned by r_admin extract

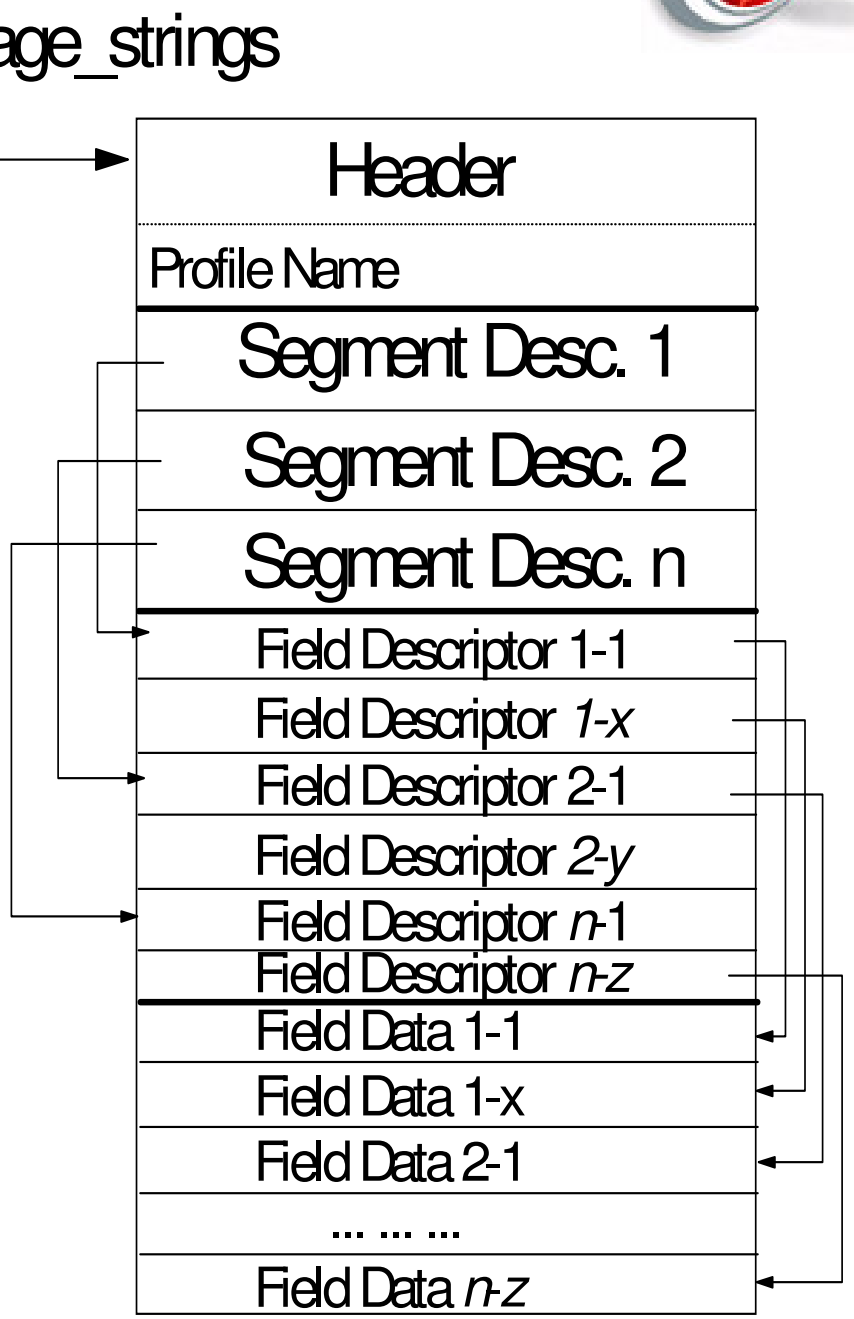
- Almost all data which can be displayed in the output of a LISTUSER/LISTGROUP commands is returned by R_admin extract. This includes almost all fields in all segments in the User and Group profiles.

What data is not returned by R_admin extract?

- Data in Reserved/unused RACF database fields
- Encrypted data such as Password..
- Fields described as ‘reserved for installation usage’
- Data which exists in a profile other than the one being extracted.
  - For example, CONNECT information between a USER and GROUP is split between the USER and GROUP profile. When extracting a USER, the portion of the CONNECT information contained in the GROUP profile is not displayed.
  - Use R_admin extract CONNECT to retrieve all CONNECT information if it is needed.

Output format

- The header contains information about how many segments were returned, and the total length of the output buffer.
- Each segment descriptor contains information about how many fields were returned in each segment and location of the field descriptors for that segment.
- The field descriptor contains field names and location of data for each field.



Input

- The input parameter list to R_admin extract is designed to match the output data format. The caller fills in the profile name and the header.

- An application using the ADMN_xxx_NEXT functions to iterate through all profiles in the database, may pass in the output of the previous extract request as input for the next request.

Input options

Options may specified in the input parameter list.

- Skipauth – A supervisor state/key 0 caller may request that no authorization checking is done to speed up the request.

- Baseonly – Any caller may request that only base segment information is returned for the specified profile. This reduces the amount of RACF database I/O and subsequent data processing.

Authorization

- Problem state caller must have READ access to FACILITY class profile IRR.RADMIN.xx, where xx=LISTUSER (user & connect) or LISTGROUP (group).
- Additionally users must have the same access as would be required to perform a LISTUSER or LISTGROUP on the profile being extracted.
- Field Level Access checking is performed on all non-base segments in the profile, if field level access checking is active on the system.

Session Summary

- New function added to R-admin to allow RACF User, Group and Connect information to be extracted easily by applications.

- This is an improvement over the complexities of using RACROUTE REQUEST=EXTRACT, or the inconvenience of parsing output from commands.

Publications

Detailed documentation for r_admin can be found in

- *z/OS Security Server RACF Callable Services - SA22-7691*

zSeries Security Update 2005

z/OS V1R7 Security Server

RACF Nested ACEE Support

Redbooks

International Technical Support Organization

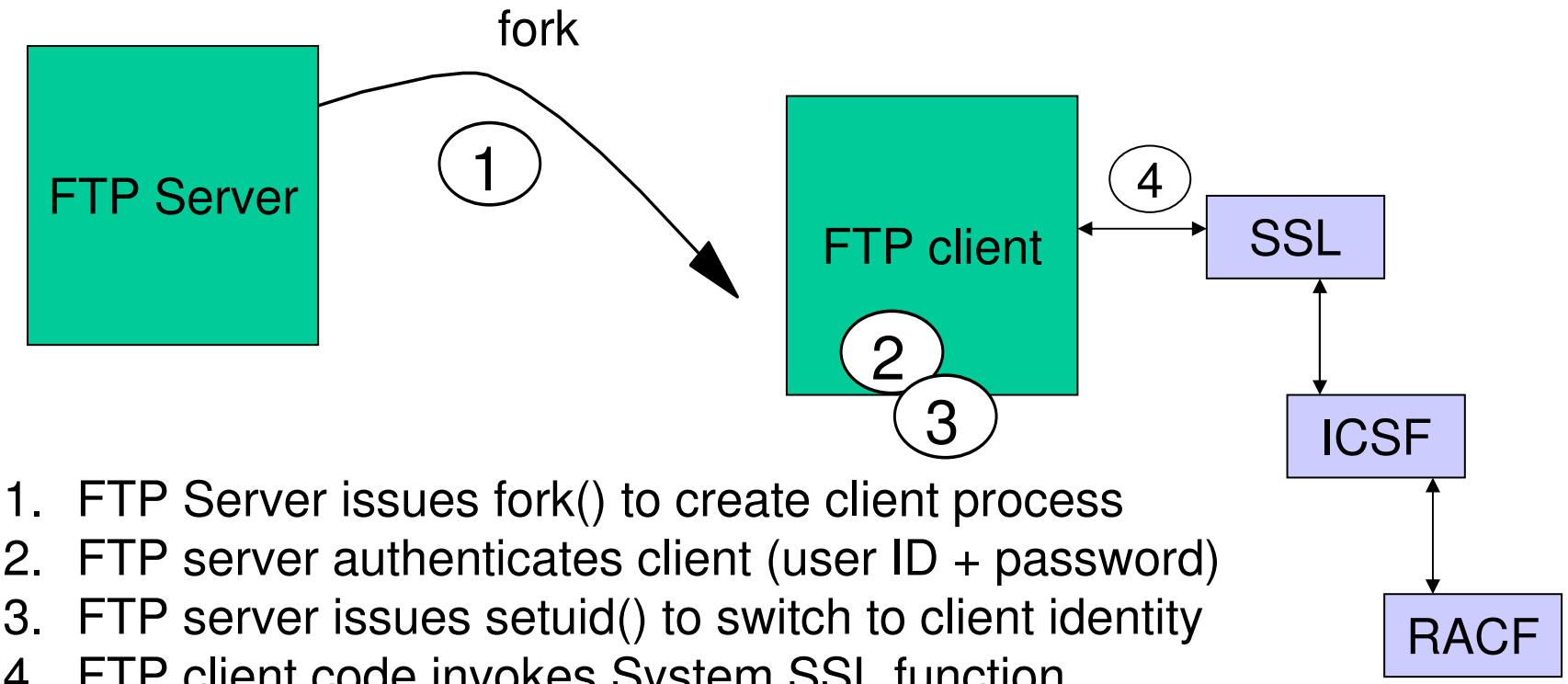
Session Objectives

- Describe new nested ACEEs exploited by FTP Server.

Problem being solved

- FTP client address space, running trusted code, needs access to protected ICSF keys and services
- Client running under identity of end-user, not server
- Client ID must be permitted to sensitive ICSF resources
- Client user ID can then access those resources in context other than as an FTP client

Big Picture



1. FTP Server issues fork() to create client process
2. FTP server authenticates client (user ID + password)
3. FTP server issues setuid() to switch to client identity
4. FTP client code invokes System SSL function
   - Which invokes ICSF function
     - Which calls RACF for authorization check

Solution

- Invent a new type of ACEE called a Nested ACEE
  - Contains embedded server identity
- Invent a new type of resource called a delegated resource
  - Honors Nested ACEEs during authorization check
- Invent a new z/OS UNIX environment variable (BPXK_DAEMON_ATTACH) which directs the setuid() family to preserve invoking identity within new Nested ACEE for client
- FTP Server code exploits the new environment variable

Value

- Problem has been addressed in service stream and prior releases as stopgap measure
- R7 solution provides a more generalized solution which does not require application-specific code
- Function could be valuable to (ported) UNIX daemons in general, not FTP in particular

Interactions & Dependencies

- z/OS UNIX provides an environment variable which uses exploit nested ACEEs (transparently) for daemon programs
  - z/OS Communication Server's FTP server exploits the environment variable

Migration/Coexistence Considerations

- There is a new message line of ICH408I
- Identifies primary user ID if access is denied to a delegated resource when a nested ACEE is present

```
ICH408I USER(IBMUSER ) GROUP(SYS1    ) NAME(SUPER DAEMON DUDE   )
PRIMARY USER(CLIENT1 )
CSFENC CL(CSFSERV )
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

Usage & Invocation
Nested ACEEs and RACOUTE

- **RACROUTE REQUEST=VERIFY**, USERID=CLIENT, PASSWRD=, … , **NESTED=YES**
  - Embeds address space identity into CLIENT's ACEE (as the 'nestling')
- With ENVROUT=
  - Creates Nested ENVR object
- With ENVRIN=
  - Creates Nested ACEE if input ENVR object is nested
- Cannot nest more than one level deep

Usage & Invocation
Nested ACEEs and RACOUTE …

- **RACROUTE REQUEST=VERIFY**, USERID=CLIENT2, PASSWRD=, … , **NESTED=COPY**
  - Embeds address space nestling into CLIENT2's ACEE

- Used by 'user' subcommand of FTP
  - Allows identity switch by specifying target user ID/password
  - COPY preserves server identity in new client

Usage & Invocation
Nested ACEEs and RACOUTE …

- **RACROUTE REQUEST=EXTRACT**,
  TYPE=ENVRXTR, ACEE=
  - Creates Nested ENVR object if ACEE is nested

Usage & Invocation
Nested ACEEs and RACOUTE …

- **RACROUTE REQUEST=FASTAUTH**, ACEE=, or, ENVRIN=
  - If client (aka 'primary') identity fails auth check, FASTAUTH re-drives auth check under embedded identity if
    - ✓Caller is supervisor state or system key
    - ✓Resource is delegated (see next slide)
- No RACROUTE REQUEST=AUTH or ck_access (z/OS UNIX file/directory) support
- Audited in a single SMF record (with both user IDs identified if nestling was used)
  - ICH408I violation messages show both identities if nestling was used
- FASTAUTH exits (ICHRFX03/04) invoked only once

Usage & Invocation
Nested ACEEs and RACOUTE and Callable Services

- RACROUTE REQUEST=SIGNON,ENVRIN= tolerates a nested ENVR object, but will ignore the nestling

- initACEE (IRRSIA00) callable service will accept a nested ENVR object as input, and will return a nested ACEE as output, but only if a managed ACEE is not requested
  - Parameter list error if ENVR object is nested

- R_fork (IRRSFK00)
  - UNIX fork processing now uses ENVR objects to copy security environment to child process

Usage & Invocation
Delegated Resources

- Defined by placing “RACF-DELEGATED” string anywhere in APPLDATA of covering profile
- By definition, only applicable to RACLISTed classes (because only supported by FASTAUTH)
- Not something you want to do unless directed to do so by your application documentation

Usage & Invocation Delegated Resources …

- MultiLevel Security (MLS) considerations
  - When SETROPTS MLACTIVE in effect, SECLABELs of primary (client) and nestling (demon) must be equivalent
    - Or SAF return code X'08', SAF reason code X'38', and RACF return code X'14' from VERIFY
  - WHEN SETROPTS SECLABELCONTROL in effect, only a system SPECIAL user can make a resource delegated when it has a SECLABEL

Messages

- ICH408I

  ICH408I USER(IBMUSER) GROUP(SYS1) NAME(SUPER DAEMON DUDE)
  PRIMARY USER(CLIENT1)
  CSFENC CL(CSFSERV)
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(READ) ACCESS ALLOWED(NONE)

- ICH10319I You are not authorized to define this resource as delegated.
  - RDEFINE
- ICH11312I You are not authorized to define this resource as delegated.
  - RALTER

RACROUTE Macro

- REQUEST=VERIFY,ENVIR=CREATE,
  NESTED=[YES|NO|COPY]

Auditing

- New SMF relocate section 390
  - Primary (client) user ID
- Existing relocates 65 and 316 – ACEE type
  - New bit for Nested ACEE
- SMF Unload Utility (IRRADU00) unloads relocate 390, and unloads the new relocate 65 bit as the string “NESTED”

Mapping Macros

- IHAACEE – ACEE
  - Adds ACEENSTA - pointer to embedded identity
  - Adds ACEENSTE – bit indicating that nestling may be used in access check
- ICHRIXP – RACROUTE VERIFY exit plist
  - Bits to indicate if NESTED=YES/COPY coded
- IRRPRIPL – RACROUTE VERIFY plist
  - Same bits as above

Session Summary

- New function solves FTP problem and provides new function for daemon based applications
- UNIX environment variable BPXK_DAEMON_ATTACH eases exploitation
- New concept: Nested ACEEs
- New concept: Delegated resources

Publications

The following RACF publications contain the details

- *z/OS Security Server RACF Command Language Reference* - *SA22-7687*
  - APPLDATA usage on RDEFINE/RALTER
- *z/OS Security Server RACROUTE Macro Reference* - *SA22-7692*
  - New **NESTED=** keyword on VERIFY
- *z/OS Security Server RACF Macros and Interfaces* - *SA22-7682*
  - New SMF relocate section
- *z/OS Security Server RACF Security Administrator's Guide* - *SA22-7683*
  - Delegated resources
- *z/OS Security Server RACF Data Areas* - *GA22-7680*
  - Mapping macro updates
- *z/OS Security Server RACF System Programmer's Guide* - *SA22-7681*
  - FASTAUTH exit considerations