IBM z Systems

RACF® for z/OS ® V2.4 Update

Bruce Wells, CISSP®

z/OS ® Security Server (RACF) Design and Development

IBM ® Poughkeepsie

RACF Users Group of New England 21 May, 2019

Session: 24318

z/OS V2.4 Preview

- *IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.*
- *Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.*
- *The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality.*
- *Information about potential future products may not be incorporated into any contract.*
- *The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.*

Trademarks:

See URL: **http://www.ibm.com/legal/copytrade.shtml** for a list of trademarks.

RACF for z/OS V2.4 Preview

- **PassTickets Enhancements**
- **Custom Fields Enhancements**
- **R_Admin & IRRXUTIL Enhancements**
- **ACEE Modification Detection**
- **Pervasive Encryption**
- **Identity Token Support**
- **IBM MFA** – Multi-Factor Authentication Support

PASSTICKET ENHANCEMENTS

PassTickets Overview

What is a PassTicket?

- A one-time-use password substitute based on a **user ID**, an **application name**, the **current time**, and a **shared symmetric key** between the application generating the PassTicket and the application (RACF) evaluating it.
- Also known as 'Secured Signon'

PassTicket Usage:

- PassTickets can be generated on z/OS or off platform – The algorithm is published
- **Applications:** Session managers, z/OS Comm Server - Express Logon Facility, WebSphere ..

Learn more:

- RACF Security Administrator's Guide – 'Using the secured signon function' section
- RACF Macros and Interfaces – 'The RACF secured signon PassTicket'

PassTickets Overview …

PassTicket Keys are defined in the SSIGNON segment of a PTKTDATA class profile:

- The **KEYMASKED** option results in a masked key stored in the RACF database. 😮
- The **KEYENCRYPTED** option results in an encrypted key stored in an ICSF key token. The generated ICSF CKDS key label is stored in the RACF database. 😀

Examples:

```
RDEFINE PTKTDATA MYAPPL SSIGNON(KEYMASKED(1234567812345678))

RDEFINE PTKTDATA MYAPPL SSIGNON(KEYENCRYPTED(1234567812345678))
```

PassTicket Keys must be protected:

- Any entity that has access to the PassTicket keys defined on the system can logon to any user who is authorized to use that application.

PassTickets – Key Label Reporting

PassTickets SSIGNON Segment Key Label reporting:

- Currently, **RLIST** reports the following information for the **SSIGNON** segment:
  - When KEYMASKED: `KEYMASKED DATA NOT DISPLAYABLE`
  - When KEYENCRYPTED: `KEYENCRYPTED DATA NOT DISPLAYABLE`

NEW In V2.4, **RLIST** will report the ICSF Key Label name:

- When KEYENCRYPTED:

Available on 2.3 with OA56831

```
KEYENCRYPTED LABEL: IRR.SSIGNON.SY1.07192018.185056.915782
```

- Additional SSIGNON Key Label reporting:
  - **DBUNLOAD**, **R_Admin** and **IRRXUTIL** are also enhanced to report the SSIGNON segment ICSF Key Label

PassTickets – KEYMASKED Migration

Available on 2.3 with OA56831

Migrate KEYMASKED to KEYENCRYPTED:

- The new **ENCRYPTKEY** keyword can be used to encrypt a **KEYMASKED** key and move it into ICSF.

```
RALTER PTKTDATA MYAPPL SSIGNON(ENCRYPTKEY)
```

- PassTicket KEYMASKED Keys can be converted in bulk with the SEARCH command:
  - Generate the CLIST:

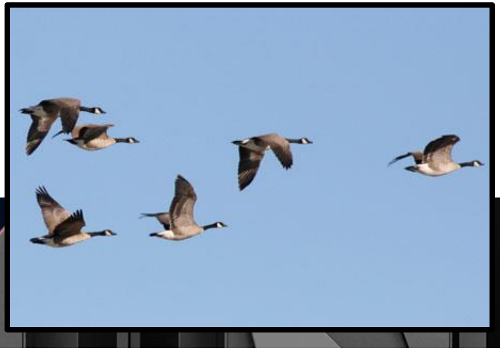
```
SEARCH CLASS(PTKTDATA) CLIST(‘RALTER PTKTDATA ‘ ‘  SSIGNON(ENCRYPTKEY)’)
```

  - Review results which are saved in the dataset:

```
‘MYUSER.EXEC.RACF.CLIST’
```

  - Run the Exec:

```
EXEC ‘MYUSER.EXEC.RACF.CLIST’
```



PassTickets – KEYLABEL Keyword

Available on 2.3 with OA56831

Creating a PassTicket Key:

- Currently the only way to create a PassTicket key is through RACF commands.

```
RDEFINE PTKTDATA MYAPPL SSIGNON(KEYMASKED(1234567812345678))

RDEFINE PTKTDATA MYAPPL SSIGNON(KEYENCRYPTED(1234567812345678))
```

NEW **KEYLABEL keyword:**

- In V2.4, the new **KEYLABEL** keyword can be used to associate an existing ICSF key with a PTKTDATA class profile:

```
RALTER PTKTDATA MYAPPL SSIGNON(KEYLABEL(IRR.PASSTICKET.MYAPPL.KEY))
```

- Allows the installation to set its own ICSF key label naming convention.
- Listing the MYAPPL application will show the updated the PassTicket key label:

```
RLIST PTKTDATA MYAPPL SSIGNON

KEYENCRYPTED LABEL: IRR.PASSTICKET.MYAPPL.KEY
```

PassTickets – Configuration Simplification

ICSF modules in LPA Requirement:

- Currently, RACF requires that a number of ICSF modules are copied into the Link Pack Area (LPA) in order to use the KEYENCRYPTED option.
- Failure to do so can result in some difficult to diagnose errors when generating or evaluating a PassTicket.

NEW In V2.4, RACF no longer requires ICSF Modules in LPA.

PassTickets - Diagnostics

PassTickets Diagnostics:

- When a PassTicket does not evaluate successfully, debugging can be difficult.

SMF Type 80 Event Code 1 (RACINIT) Relocate 443:

- When MFA support was added to RACF, relocate section 443 was added to all of the SMF type 80 event code 1 records to indicate details about the authentication attempt.
- Relocate 443 indicates whether a PassTicket was used for authentication and whether it was successful.
    - When not successful, the record includes internal return and reason codes

  In V2.4, Relocate 443 will also include:

    - New reason code which indicates how far the PassTicket was from validity (time offset)
    - The application name used in the evaluation process (caller provided or derived internally)

RCVTPTGN:

- The RCVT anchored PassTicket generation service can be difficult to debug

  In V2.4, Register 0 will contain a failure Reason Code

CUSTOM FIELDS ENHANCEMENTS

Custom Fields Overview

- Custom fields are fields within the RACF database that you customize to store security information about the **users** and **groups** at your installation.

- You can tailor the names and attributes of custom fields.

- Once you define custom fields, use RACF commands, such as the **ALTUSER** and **ALTGROUP** to add data to custom fields.

- Fields are stored in the **CSDATA** segment

Custom Fields Example

Define a new USER class field for the employee Serial Number called EMPSER:

```
RDEFINE CFIELD USER.CSDATA.EMPSER UACC(NONE)
     CFDEF(TYPE(NUM) FIRST(NUMERIC) OTHER(NUMERIC) MAXLENGTH(8)
           MINVALUE(100000) MAXVALUE(99999999)
           HELP(’EMPLOYEE SERIAL NUMBER, 6 – 8 DIGITS’) LISTHEAD(’EMPLOYEE SERIAL=’))
```

Activate the CFIELD class:

```
SETR CLASSACT(CFIELD)
```

Update RACF command dynamic parse:

```
IRRDPI00 UPDATE
```

Use the custom field to assign a user a Serial Number:

```
ALTUSER COOP CSDATA(EMPSER(123456))
```

List the custom field:

```
LISTUSER COOP CSDATA NORACF
USER=COOP
CSDATA INFORMATION
----------------------------------------
EMPLOYEE SERIAL= 123456
```

What about General Resource and Dataset profiles?!?!

Custom Fields – General Resource and Dataset Profiles

In V2.4, Custom Fields will support General Resource and Dataset class profiles!

- Works in the a consistent fashion with the existing ability for user and group profiles.

- For example, to create a DATASET profile field to contain character data:

```
RDEFINE CFIELD DATASET.CSDATA.MYFIELD
               CFDEF(TYPE(CHAR) MAXLENGTH(50))
```

- To create a similar General Resource field:

```
RDEFINE CFIELD GENERAL.CSDATA.MYFIELD
               CFDEF(TYPE(CHAR) MAXLENGTH(50))
```

- Note that a general resource field will apply to any general resource class by default.
  - You can write an exit to restrict the field to certain resource classes.

Custom Fields – Validation Exit

NEW New VALREXX keyword:

- Identifies a REXX exec to be used to validate the field.
- Supported by all class types (USER, GROUP, DATASET and General Resource)
- Same exit can optionally be shared by multiple fields.

- Example VALREXX:

```
  RALTER CFIELD DATASET.CSDATA.MYFIELD
                       CFDEF(VALREXX(VALMYFLD))
```

- **Note:** The existing IRRVAF01 dynamic exit is also supported for the DATASET and general resource fields.

R_ADMIN & IRRXUTIL ENHANCEMENTS

R_Admin Overview

R_Admin Callable service:

- RACF/SAF callable service which provides a programming interface to perform RACF administrative functions and retrieve RACF security data.
  - **Run RACF Commands**
  - **Retrieve Security Configuration:**
    1. RACF Profiles: USER, GROUP, General Resource classes **(*not DATASET class profiles*)**
    2. SETROPTS Configuration
    3. RACF Remote Sharing (RRSF) Configuration information
  - **Update Security Configuration:**
    1. RACF Profiles: USER, GROUP, General Resource classes, DATASET profiles
    2. SETROPTS Configuration

R_Admin Enhancements

R_Admin Callable service:

- RACF/SAF callable service which provides an API to perform RACF administrative functions and retrieve RACF security data.
    - **Run RACF Commands**
    - **Retrieve Security Configuration:**
        1. RACF Profiles: USER, GROUP, General Resource classes, **DATASET** ~~(*not DATASET class profiles*)~~
        2. SETROPTS Configuration
        3. RACF Remote Sharing (RRSF) Configuration information
    - **Update Security Configuration:**
        1. RACF Profiles: USER, GROUP, General Resource classes, DATASET profiles
        2. SETROPTS Configuration

NEW In V2.4, R_Admin can retrieve DATASET class profile fields!

Authority Required: READ access to IRR.RADMIN.LISTDSD in the FACILITY class

IRRXUTIL Overview

IRRXUTIL is a program that creates a set of REXX stem variables for several categories of RACF information.

1. **RACF Profiles:**

   USER, GROUP, General Resource classes, **(not DATASET profiles)**

2. **SETROPTS Configuration**
3. **RACF Remote Sharing (RRSF)** Configuration information

IRRXUTIL Enhancements

IRRXUTIL is a program that creates a set of REXX stem variables for several categories of RACF information.

1. **RACF Profiles:**

   USER, GROUP, General Resource classes, **DATASET**

2. **SETROPTS Configuration**
3. **RACF Remote Sharing (RRSF)** Configuration information
4. **Class Descriptor Table (CDT) Entries**

NEW In V2.4, IRRXUTIL can retrieve DATASET class profiles!

NEW In V2.4, IRRXUTIL can retrieve CDT entries!

IRRXUTIL Enhancements – CDT Information

NEW **IRRXUTIL can retrieve Class Descriptor Table entries:**

- Supports both static and dynamic classes
  - CDT information is obtained with RACROUTE REQ=STAT, for the named class or the one following it
- The current SETROPTS settings for the class can optionally be returned
  - SETROPTS class settings are obtained with R_Admin

Example XFACILIT Class CDT Info:

```
CLS.CLASSNAME=XFACILIT
CLS.ID=1
CLS.POSIT=8
CLS.MAXLNTH=246
CLS.OPERATIONS=0
CLS.RACLIST_ALLOWED=1
CLS.RACLIST_REQUIRED=0
CLS.GENERIC_ALLOWED=1
…
```

Example SETROPTS Class Data:

```
CLS.RACLIST_ACTIVE=1
CLS.STATISTICS_ACTIVE=0
CLS.GENERIC_ACTIVE=1
CLS.GENCMD_ACTIVE=1
CLS.GENLIST_ACTIVE=0
CLS.GLOBAL_ACTIVE=0
CLS.LOGOPTIONS=DEFAULT
CLS.AUDIT_ACTIVE=1
…
```

ACEE MODIFICATION DETECTION

ACEE Overview

- **The ACcessor Environment Element (ACEE) is a control block to represent a user’s security environment.**
  - The ACEE is created by RACF/SAF when a user authenticates to a z/OS application.
- **The contents are derived from information in the USER profile, containing:**
  - The user ID
  - List of GROUPs
  - Various authorities (SPECIAL, AUDITOR, OPERATIONS, etc)
  - Lots of other security environment details
- **The ACEE is used by RACF commands and RACF authorization checking to determine authority and access to resources.**
  - Does this USER have the authority required to perform this action (SPECIAL, AUDITOR…)
  - Does this USER have this level of access to this RESOURCE in this CLASS of resources.
- **It is anchored in the address space, or task, or created by an authorized application and passed explicitly to various SAF services.**

ACEE Modification Detection

NEW **In V2.4, RACF can detect changes to a user's ACEE that result in elevated privilege:**

- A new message is issued when such a modification is detected.
- Exceptions can be defined for trusted applications in order to suppress the message for users of such an application.

Fingerprint:

- New fingerprint field in ACEE is created with RACROUTE REQ=VERIFY
- Fingerprint encapsulates the User ID and various authority-related fields (SPECIAL, TRUSTED…)

Value:

- Useful in detecting programs that fall outside your security policy
- Useful in detecting programs that might be requesting more privilege than absolutely necessary

ACEE Modification Detection

What type of applications 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

ACEE Modification Detection - Configuration

ACEECHK Class:

- ACEE fingerprint is verified when **ACEECHK** class is active
- New message **IRR421I** is issued when privilege escalation is detected

```
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
```

Program Exceptions:

- **IRR421I** can be suppressed for trusted programs by defining exception profiles in the **ACEECHK** class

```
RDEFINE ACEECHK IRR.EXCLUDE.TESTPROG
```

- When **IRR.ABEND.ON.FAILURE** is defined in the **ACEECHK** class:
  - When privilege escalation is detected and no exception is defined then **ABEND 4C6** is issued with new reason code **27CC**(**X’ACE**’).

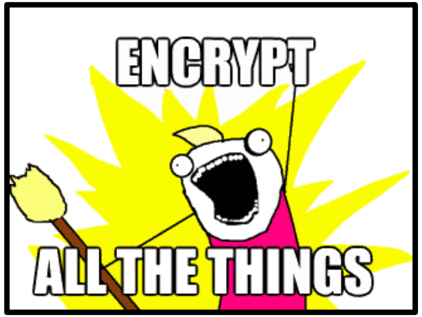
```
IRR421I ACEE modification detected
  for user IBMUSER in address space ID 0x0000001B running under
  user BCSCGB1 and job name CKFCOLL1 while program DSN3ID00 is
  running. The RACF function detecting the modification is IRRRCK00.
  Rsn=0x40008000.  (ACEEPRIV is ON) (ACEEUSRI: expected BCSCGB1,
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

PERVASIVE ENCRYPTION SUPPORT

Pervasive Encryption – JES Spool Data Sets

NEW **JES segment, intended for profiles in the JESJOBS class, containing a KEYLABEL field.**

- As with data set (pervasive) encryption, the label refers to the ICSF encryption key to be used while encrypting JES spool data.



RALTER

JES | NOJES

JES

Specifies the JES information for the profile being changed.

KEYLABEL(key-label) | **NOKEYLABEL**

KEYLABEL(key-label)

Specifies the name of an ICSF key label to be used when encrypting spool data for resources that are covered by the profile.

NOKEYLABEL

Specifies that you want to delete the key-label from the JES segment of the profile.

NOJES

Specifies that you want to delete the JES segment from the profile.

IDENTITY TOKEN SUPPORT

Identities in RACF

Identities in z/OS can be assigned:

- **Directly** with the presentation of a user ID and an authenticator
  - TSO logon, batch job with user ID and password/password phrase
- **By inheritance**
  - Submitting a batch job from an authenticated session
- **By assertion**, as in the authentication is done outside of RACF and the authenticated identity is trusted
  - Resource manager specification of target identity
  - Surrogate job submission
  - PASSTICKETs
- **By mapping**, “Mapped”, were an external identity is asserted and then is mapped to a z/OS identity

NEW

With z/OS V2.4, RACF is introducing a new assertion mechanism, the JSON web token (pronounced “jot”)

Identity Token Support Overview

Identity Token:

- An Identity Token is used to assert user claims which can be trusted by the consumer of the token.
- Our use adheres to the JSON Web Token (JWT) IETF specifications: RFC 7519

RACROUTE Support for Identity Tokens: RACROUTE authentication processing can generate and validate Identity Tokens (IDT).

- **Generation:** Applications can request that an IDT be returned from RACROUTE.
- **Validation:** Applications can supply an IDT to authenticate a user instead of other credentials.

IDT Configuration:

- The security administrator can create profiles in the IDTDATA class:
  - Configure how certain fields in an IDT are generated and validated

Identity Token Support Overview

Linking Multiple Authentication API Calls:

- In some cases, user authentication requires multiple steps:
  - **Expired Password / Invalid New Password / MFA Expired PIN …**
- **Problem:**
  - MFA credentials are one time use.
  - When multiple authentication calls are required, an already consumed MFA token will fail.
- **Solution:**
  - The Identity Token can be used to link authentication status information between multiple authentication API calls without replaying the MFA credentials.

Identity Token Support Overview

Replaying Proof of Authentication:

- Some applications authenticate a user and “replay” that authentication multiple times.
- **Problem:**
  - Some applications cache the user provided credential and replay it back again later.
  - For users with one time use MFA tokens, this does not work.
- **Solution:**
  - The Identity Token support allows applications to authenticate a user and receive proof of that authentication which can be supplied back to RACROUTE in place of other credentials like a password.
  - Signed JWTs can be returned to an end user for later use by the application.

JWT – JSON Web Token

A JSON Web Token (JWT) is used to assert claims between multiple parties. They are often used to prove a user has been authenticated.

- **JWT RFC7519:** https://tools.ietf.org/html/rfc7519

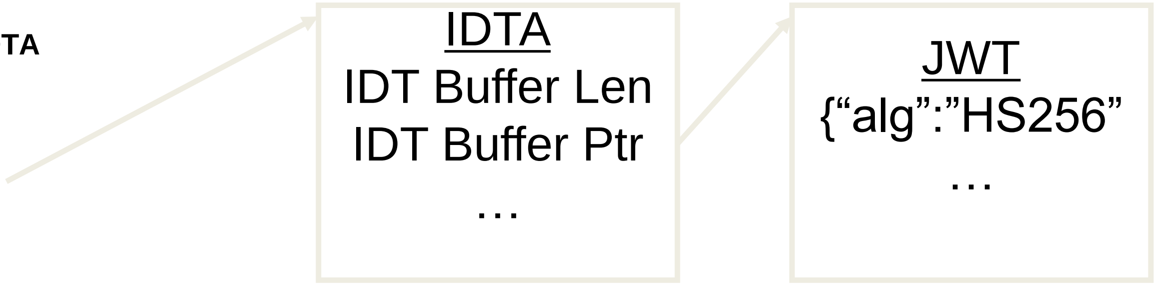
JWT:

- **Header (JOSE):**
  - **{“alg” : “HS256” or “none”}** – Signature Algorithm: **HS256** = HMAC with **SHA-256**, none = unsecured
- **Body Claims – (JWS Payload):**
  - **{“jti” : “cb05…”,** – JWT Unique identifier
  - **“iss” : “saf”,** – Issuer name – Entity that created the JWT
  - **“sub” : “USER01”,** – Subject (the authenticated user)
  - **“aud” : “CICSLP8”,** – Audience – Target consumer of the JWT
  - **“exp” : 1486744112,** – Expiration time - (Seconds since 1970 - Expired tokens should be rejected)
  - **“iat”: 1486740112,** – Issued at – The time at which the JWT was issued.
  - **“amr”:[”mfa-comp”,”saf-pwd”]}** – Authentication Method References - Indicates how the subject was authenticated
- **Signature (JWS)** – Encoded in Binary
  - 389A21CD32108C3483DA

Identity Token Externals – RACROUTE REQ=VERIFY

- **New RACINIT Parameter: IDTA**

```
RACROUTE REQUEST=VERIFY
     ,...
     ,IDTA=idta_data_addr
     ,RELEASE=PLV0001
     ,...
```



IDTA - Specifies the address of the data structure that describes the identity token data. The address points to a data structure defined in a new SAF mapping macro named IRRPIDTA. The IDTA keyword can only be specified when RELEASE is set to PLV0001 or higher.

Identity Token Externals – RACROUTE REQ=VERIFY, RELEASE=

- **RELEASE**=number

specifies the release level of the parameter list to be generated by this macro. Through RACF 2.2, it corresponds to the FMID of the RACF release. After that, when RACF became solely an element of OS/390® or z/OS, it corresponds to the FMID of the RACF.

NEW: Starting with HRF77C0, the naming convention for the RELEASE keyword is updated to correspond to a parameter list version number. Version PLV0001 is the initial parameter list version number and contains all parameters in HRF77C0 and earlier.

…

77A0 corresponds to FMID HRF77A0 (z/OS Security Server V2R2)

77B0 corresponds to FMID HRF77B0 (z/OS Security Server V2R3)

PLV0001 corresponds to FMID HRF77C0 (z/OS Security Server V2R4)

Administrative Control over IDTs

IDTDATA Class profiles and IDTPARMS segment:

- Security administrators can control the use of tokens by defining profiles in the new IDTDATA general resource class, using a new IDTPARMS segment

IDTDATA class:

- Must be **ACTIVE** before Identity Tokens will be generated or validated
- Must be **RACLISTed** before any profiles in the class will be used

IDTDATA profile format: <IDT Type>.<application name>.<user ID>.<IDT issuer name>

  - IDT Type – "JWT"
  - Application name – The value specified in the APPL= parameter
  - User ID – the user being authenticated
  - IDT issuer name – "SAF"

Note: Generics are allowed. When a user is authenticated with a JWT, the best matching profile is used.

Administrative Control over IDTs …

IDTPARMS segment RALTER command keywords

```
[ IDTPARMS(
    [ SIGTOKEN(pkcs11-token-name)  | NOSIGTOKEN ]
    [ SIGSEQNUM(pkcs11-sequence-number) | NOSIGSEQ ]
    [ SIGCAT(pkcs11-category) | NOSIGCAT ]
    [ SIGALG( HS256 | HS384 | HS512 ) | NOSIGALG ]
    [ ANYAPPL(YES | NO) ]
    [ TIMEOUT(timeout-minutes) ]
)
NOIDTPARMS ]
```

Administrative Control over IDTs …

IDTPARMS segment RALTER command keywords

```
[ IDTPARMS(
     [ SIGTOKEN(pkcs11-token-name)  | NOSIGTOKEN ]
     [ SIGSEQNUM(pkcs11-sequence-number) | NOSIGSEQ ]
     [ SIGCAT(pkcs11-category) | NOSIGCAT ]
     [ SIGALG( HS256 | HS384 | HS512 ) | NOSIGALG ]
     [ ANYAPPL(YES | NO) ]
     [ TIMEOUT(timeout-minutes) ]
)
NOIDTPARMS ]
```

Location of the signing key

Signature algorithm to use

Whether token can be used by other applications

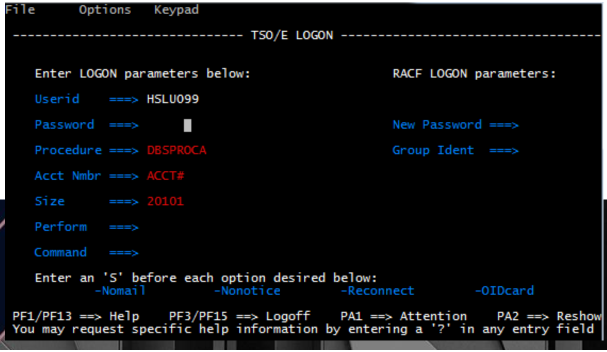
Validity interval of a token

TSO Exploitation of Identity Tokens

- TSO Logon process is updated to specify the new RACROUTE IDTA parameter.
  - Supported in both pre-prompt and normal logon screens.

- **Improves logon experience for IBM MFA users:**
  - When multiple authentication API calls are required, the Identity Token keeps track of the current authentication state.
  - **Scenarios:**
    - Expired MFA PIN or expired Password, RSA Next Token Code Mode and MFA protocols which required multiple steps.

Note: Support is not activated in RACF until the IDTDATA class is ACTIVE.



```
File      Options   Keypad

 ----------------------------- TSO/E LOGON ----------------------------------

   Enter LOGON parameters below:                      RACF LOGON parameters:

   Userid    ===> HSLU099

   Password  ===>      █                              New Password ===>

   Procedure ===> DBSPROCA                            Group Ident  ===>

   Acct Nmbr ===> ACCT#

   Size      ===> 20101

   Perform   ===>

   Command   ===>

   Enter an 'S' before each option desired below:
           -Nomail         -Nonotice        -Reconnect        -OIDcard

 PF1/PF13 ==> Help    PF3/PF15 ==> Logoff    PA1 ==> Attention    PA2 ==> Reshow
 You may request specific help information by entering a '?' in any entry field
```

BONUS TRACKS

Password phrase support for MCS console logon

- Available now with OA54790 on z/OS V2R2+
- Enabled by defining MVS.CONSOLE.PASSWORDPHRASE.CHECK in the OPERCMDS class
- Maximum length of 45 characters
- No associated RACF APAR

Log stream write-only access

- Available now with OA56050 on z/OS V2R3
- Existing controls grant ability to read, write, and delete data from a log stream
  - UPDATE access grants all three abilities
- However, it may be desirable to only allow writing
  - E.g. a trace log or other type of collection to which multiple sources will contribute
- Now, profiles can be created to simply allow write access

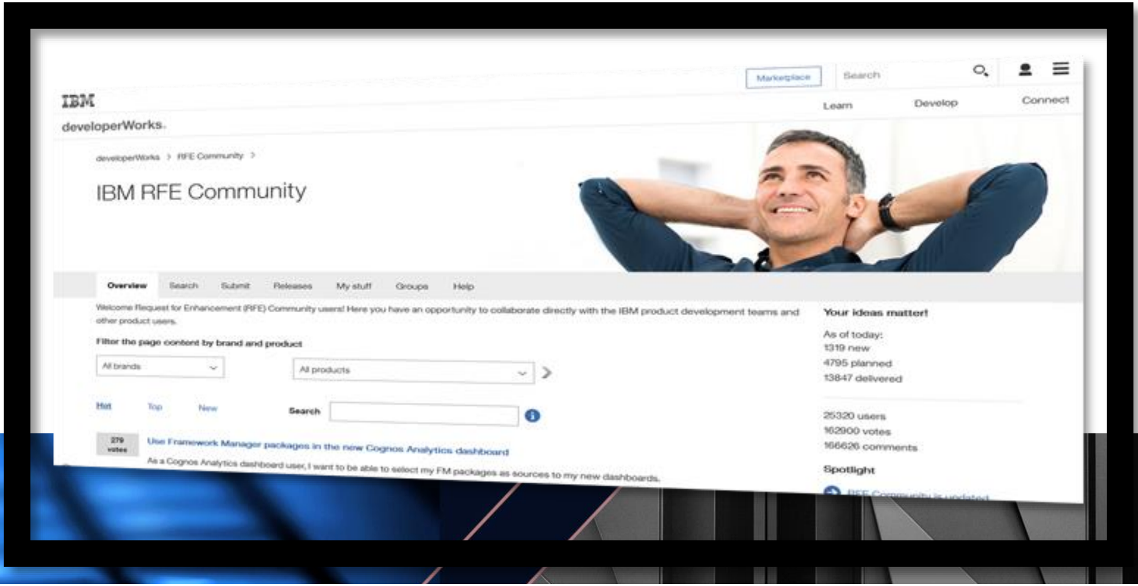
```
RDEFINE LOGSTRM WRITE_ONLY_log-stream-name
PERMIT WRITE_ONLY_log-stream-name CLASS(LOGSTRM) ID(APP1) ACCESS(UPDATE)
```

- Corresponding RACF APAR OA57159 increases the maximum length of profiles in the LOGSTRM class

REQUEST FOR ENHANCEMENTS (RFE)

Request For Enhancements (RFE)

- **Requirements should be submitted to IBM via RFE:**
  - Reviewed by the design and development team
  - Facilitates a dialog between clients and IBM
  - **Link:** https://www.ibm.com/developerworks/rfe



IBM RFE Community

Welcome Request for Enhancement (RFE) Community users! Here you have an opportunity to collaborate directly with the IBM product development teams and other product users.

Filter the page content by brand and product

279 votes Use Framework Manager packages in the new Cognos Analytics dashboard

As a Cognos Analytics dashboard user, I want to be able to select my FM packages as sources to my new dashboards.

Your ideas matter!

As of today:
1319 new
4795 planned
13847 delivered

25320 users
162900 votes
166626 comments

Spotlight

RFE Community is updated

MULTI-FACTOR AUTHENTICATION

Black Hat 2017 Hacker Survey Report[1]

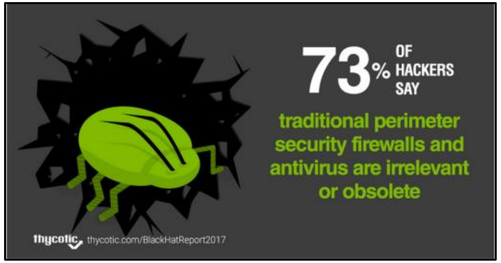
QUESTION: What type of security is the hardest to get past?

68% say multi-factor authentication and encryption are biggest hacker obstacles

32% OF HACKERS SAY

accessing privileged accounts was the number one choice for the easiest and fastest way to get at sensitive data

thycotic thycotic.com/BlackHatReport2017



73% OF HACKERS SAY

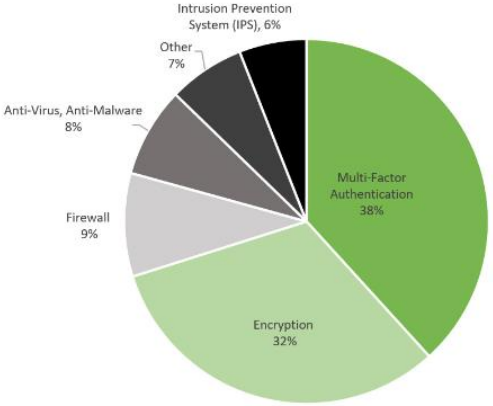
traditional perimeter security firewalls and antivirus are irrelevant or obsolete

thycotic thycotic.com/BlackHatReport2017

80% OF HACKERS SAY

humans are the most responsible for security breaches

thycotic thycotic.com/BlackHatReport2017



[1] thycotic Black Hat 2017 Hacker Survey Report https://thycotic.com/resources/black-hat-2017-survey/

Compliance

<u>PCI DSS v3.2</u>

8.3 Secure all individual non-console administrative access and all remote access to the CDE using multi-factor authentication.

8.3.1 Incorporate multi-factor authentication for all non-console access into the Cardholder Data Environment (CDE) for personnel with administrative access.

Note: This is a best practice until January 31, 2018, after which it becomes a requirement.

<u>NIST SP 800-171</u>

3.5.3 Use multifactor authentication for local and network access to privileged accounts and for network access to non-privileged accounts.

Note: Network access is any access to an information system by a user (or a process acting on behalf of a user) communicating through a network (e.g., local area network, wide area network, Internet).

Note: This requirement is effective December 31, 2017.

How are users authenticating without MFA?

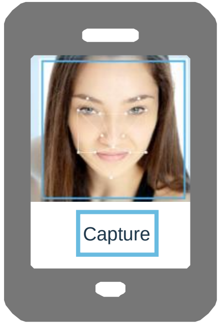
Users authenticate with:

- Passwords
- Password phrases
- Digital Certificates
- via Kerberos

Problems with passwords:

- Common passwords
- Employees are selling their passwords
- Password reuse
- People write down passwords
- Malware
- Key log
- Password cracking

What is multi-factor authentication?



Knowledge Factors
- Usernames and passwords
- PIN Code

Possession Factors
- ID Badge
- One time passwords
  - Time-based

Inherent Factors
- Biometrics
- Keyboard dynamics

IBM Multi-Factor Authentication for z/OS

Higher assurance authentication for IBM z/OS systems that use RACF

IBM Multi-Factor Authentication on z/OS provides a way to raise the assurance level of z/OS, applications, and hosting environments by extending RACF to authenticate users with multiple factors.

Fast, flexible, deeply integrated, easy to deploy, easy to manage, and easy to use

PCI-DSS
Achieve regulatory compliance, reduce risk to critical applications and data

Architecture supports multiple third-party authentication systems at the same time

IBM Multi-Factor Authentication for z/OS (5655-162)
IBM Multi-Factor Authentication for z/OS S&S (5655-163)

THE END

RACF® for z/OS ® V2.4 Update

Bruce Wells, CISSP®

z/OS ® Security Server (RACF) Design and Development

IBM ® Poughkeepsie

RACF Users Group of New England 21 May, 2019

Session: 24318