RACF® Update
z/OS 3.2 and more!

Elijah Swift, CISSP® - IBM z/OS Secure Engineering

Agenda

z/OS 3.2 – Continuous Delivery

- Password Phrase Self-Serve Provisioning
- Optional Phrase Syntax Rules

z/OS 3.2 – Base:

- RACDCERT Support for multiple SANs
- Granular Dataset Encryption
- User Quarantine

z/OS 3.1 – Continuous Delivery:

- RACF Identity Token enhancements (APAR OA65299)

RACF PASSWORD PHRASE SELF-SERVE PROVISIONING

Background

Users can be authenticated to z/OS applications using SAF/RACF authentication APIs with passwords, passwords phrases, PassTickets or MFA.

- Passwords are 1-8 characters long
- Phrases are 9-100 characters long

Migrating from passwords to phrases:

- Currently only an administrator can assign a user their initial password phrase.

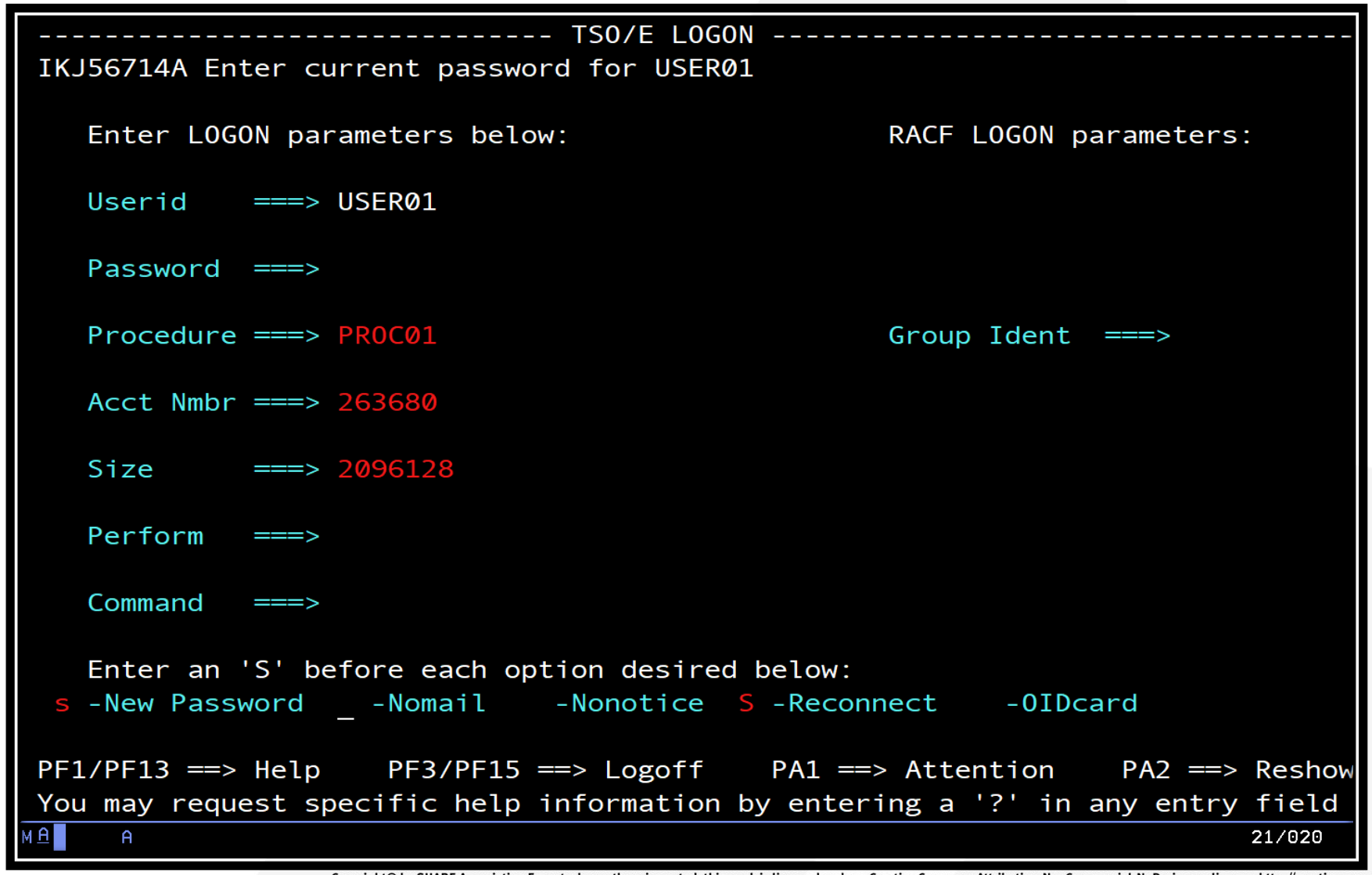
RACF Password Phrase Self-Serve Provisioning

- Currently security administrator does the work
  - Derive a secure initial password phrase value
  - Securely communicate it to large numbers of users
  - Eventually delete phrase
- Let users do it themselves
- Less work for administrators, easy for users

How do I turn it on?

- A new **IRR.PHRASE.SELF.SERVICE** resource is defined in the **XFACILIT** class
- RACROUTE REQUEST=VERIFY checks access to the new resource when PASSWRD= and NEWPHRASE= are specified and
    - the password is not a PassTicket
    - the user does not currently have a phrase
- **Defined access levels:**
    - **NONE:** The user is not allowed to establish an initial password phrase. (current behavior)
    - **READ:** The user is allowed to establish a new phrase, and RACF removes the user's password automatically. Assign this level when you are confident the user will not encounter any phrase-related application issues. This eliminates the need for an administrator to delete the password in the future.
    - **UPDATE:** The user is allowed to establish a new phrase, and the password is **not** deleted. Use this when you want the user to retain their password for a transition period, in case they encounter an application that does not support phrases. The administrator is responsible for deleting the password in the future.

Example App



- Users can change their own password to a phrase via application logon screens if the application allows the combination of a “current password” and “new phrase”.
- TSO supports phrase self-provisioning.
- This works on demand or when the password expires when an administrator grants the user access to the new resource.

SMF Records

- SMF 80 Event Code 1 (JOBINIT) Relocate Section 443 (authentication information) defines three new bits
    - A new password was accepted
    - A new phrase is accepted
    - The user's password was deleted
- SMF Unload support
- When SETROPTS AUDIT(USER) is active, password/phrase changes are logged (current behavior).
    - The fact that a 'password' change occurred has to be inferred from the logging reason, and the changed authenticator (password or phrase) has to be inferred by inspecting other bits, but it is now made explicit.

APAR Information

This support is available via continuous delivery on z/OS 3.1 and up via the PTFs for APAR:

RACF: OA68301

RACF OPTIONAL PHRASE SYNTAX RULES

Background

RACF has a set of built-in new password phrase rules which can not be disabled.

The new password phrase exit does not get control when a new phrase violates these rules.

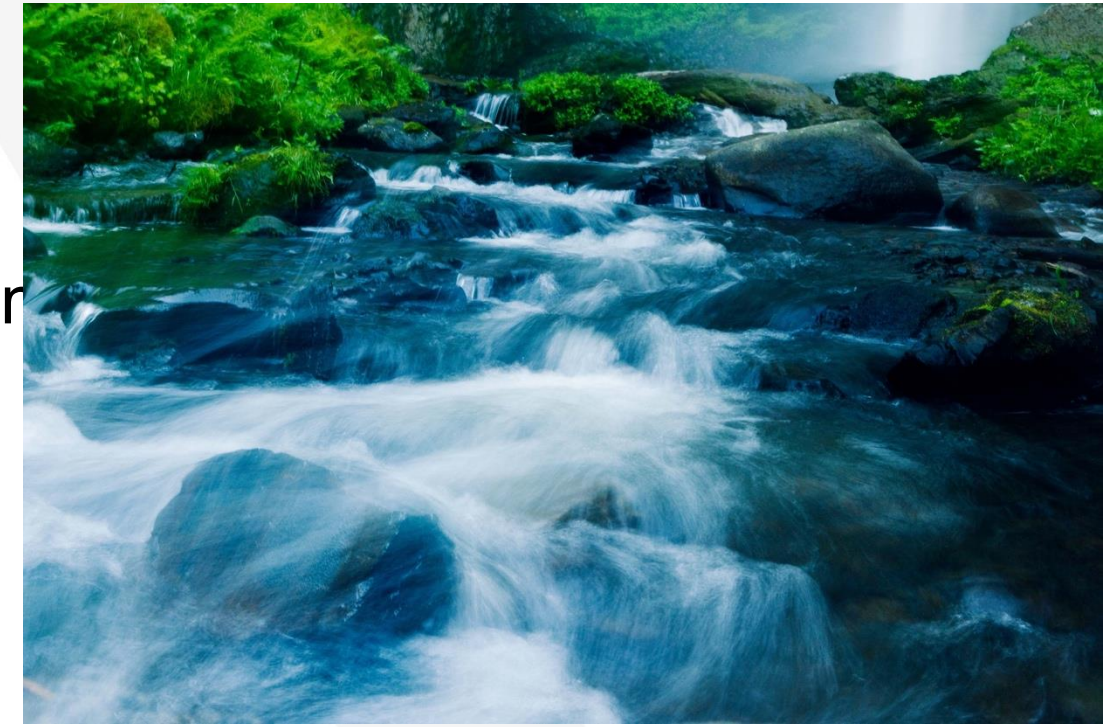
This prevents some new phrases which are valid on other platforms from being used in RACF.

Built-in password rules:

- Must not contain the user's RACF User ID (upper case or lower case)
- Must contain at least 2 alphabetic characters
- Must contain at least 2 non-alphabetic characters (numerics, punctuation, special characters, blanks)
- Must not repeat the same character consecutively more than 2 times

Optional Password Phrase Syntax Rules

- Difficult to derive a common set of long password rules that can be **consistently** enforced across the enterprise
- Allow admins to bypass the RACF built-in phrase syntax rules
- Keeps end-user experience more consistent



How do I turn it on?

- **Enablement:**
  - Define a new resource in the new OPTRACF class and RACLIST REFRESH:

```
RDEFINE OPTRACF PHRASE.BYPASS.BUILTIN.SYNTAX.RULES
SETROPTS CLASSACT(OPTRACF) RACLIST(OPTRACF)
```

- **Usage:**
  - When a user attempts to set a new password phrase, if the resource is defined, the default rules are completely bypassed

- **Updated New Phrase Sample:**
  - The Github sample is updated to contain checks for each of the (default) syntax rules so that desirable ones can be preserved

Details

- **Implementation Details:**
  - A new OPTRACF class (similar to OPTAUDIT) in which non-logging-related RACF switch profiles can be defined.
  - The ENF 62 (RACLIST) listener in the RACF subsystem address space is tweaked to look for this new class/profile pair, and set a new RCVT bit as appropriate
  - The RACF subsystem address space must be active at the time the OPTRACF class is /NO/RACLIST/REFRESHed.
  - If you wish to preserve a subset of the rules, **you must have a new phrase exit**
  - The Github sample is updated with each of the syntax rules, enabled by default

Exits and Samples

RACF Password Phrase exit (ICHPWX11):

- ICHPWX11 – New-password-phrase exit
  - Sample in samplib – Calls IRRPHREX
  - No update to ICHPWX11

REXX Sample (IRRPHREX):

- Sample in samplib, but an updated version is in GitHub
- Github sample IRRPHREX is now at Version “V6”

  https://github.com/IBM/IBM-Z-zOS/tree/main/zOS-RACF/Downloads/RexxPwExit

  Can be used to selectively enforce the same “built-in” new-phrase rules (that are now able to be bypassed).

APAR Information

This support is available via continuous delivery on z/OS 3.1 and up via the PTFs for APARs:

RACF: OA67750
- **HRF77E0:** UJ98388
- **HRF77F0:** UJ98390

SAF: OA67751
- **HBB77E0:** UJ98374
- **HBB77F0:** UJ98375

RACDCERT MULTIPLE SANS

Background on Digital Certificates

SHARE
EDUCATE • NETWORK • INFLUENCE

Digital Certificates:

- Digital documents used to bind an end entity (server or person) to a public key
- Signed by a trusted third party called a Certificate Authority (CA)
- Used to authenticate, exchange keys and sign and encrypt other digital assets.

SAN Extension:

- Can include a Subject Alternative Name (SAN) extension
  - Associate multiple different internet names with a single logical entity/server

RACF Digital Certificate Support

SHARE
EDUCATE • NETWORK • INFLUENCE

SAF/RACF Certificate Support:

- **RACDCERT** command helps you generate, store and organize certificates
- Applications get programmatic access to SAF/RACF certificates via System SSL, AT-TLS and SAF/RACF callable services (R_datalib)
- Use **GENCERT** keyword on **RACDCERT** command to generate certificates

Prior to z/OS 3.2:

- **RACDCERT GENCERT** only generate certificates with one of each type of ALTNAME
  - IP, Domain, Email, URI
- **RACDERT LIST** only lists the first of each type of **ALTNAME**

RACDCERT GENCERT Multiple SANs

SHARE
EDUCATE • NETWORK • INFLUENCE

Starting in z/OS 3.2:

- **RACDCERT GENCERT** generates certificates containing multiple SANs of each ALTNAME type
- **RACDCERT LIST** can display all the SANs contained within a certificate
- Authority key ID and subject key are also now listed

RACDCERT GENCERT Multiple SANs

RACDCERT GENCERT ALTNAME Syntax update:

Existing keywords (unchanged): ALTNAME(IP, DOMAIN, EMAIL, URI)

New keywords: ALTNAME(AIP, ADOMAIN, AEMAIL, AURI)

Allow multiple values to be specified.

```
RACDCERT GENCERT
   ...
    [ALTNAME(
       IP(numeric-IP-address)
       AIP(numeric-IP-address1,numeric-IP-address2,...)
       DOMAIN('internet-domain-name')
       ADOMAIN('internet-domain-name1','internet-domain-name2',...)
       EMAIL('email-address')
       AEMAIL('email-address1','email-address2',...)
       URI('universal-resource-identifier')
       AURI('universal-resource-identifier1','universal-resource-
identifier2',...)
    )]
```

RACDCERT GENCERT Multiple SANs


```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN = BobGens1
        Validity
            Not Before: Oct 24 05:00:00 2024 GMT
            Not After : Oct 25 04:59:59 2025 GMT
        Subject: CN = BobGens1
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:e2:c4:88:07:72:25:4c:3b:b1:ab:f9:6c:0e:7b:
                    f4:99:4e:a9:12:e4:ca:45:2c:8f:8d:16:12:11:f3:
                    40:a3:7c:a3:16:b6:6f:ea:19:d8:96:2f:f4:59:6d:
                    cf:72:fc:20:43:a0:d8:0b:02:1c:59:68:25:0d:dc:
                    4e:e9:8a:42:b8:1b:af:1e:da:e6:60:64:74:e0:0c:
                    a5:fd:30:04:5e:bd:4a:1c:33:19:47:96:14:21:e6:
                    15:06:04:5f:81:e1:85:dd:36:a1:6c:49:7e:2a:b4:
                    32:d9:62:d4:3f:eb:07:fe:3b:1a:0f:48:65:31:9b:
                    a2:83:ab:5a:32:ca:3e:26:24:d8:ca:97:ff:21:43:
                    db:92:62:cd:d3:dd:31:37:f6:db:4c:f5:8d:44:08:
                    e4:67:5c:a3:b2:3b:e2:da:c6:3e:29:e7:a8:41:bb:
                    22:ee:b7:14:bb:42:79:f2:38:2a:3d:b3:b2:26:93:
                    fb:10:7b:c2:73:77:1e:e7:05:84:e1:e4:bc:a4:ec:
                    b3:48:25:34:bb:4c:f2:49:90:c6:7a:81:8c:a1:83:
                    c8:de:a6:c5:93:14:a7:a1:33:bc:53:ae:96:92:0b:
                    94:f9:8b:48:fa:d7:ad:d5:6e:13:38:4d:b3:58:dc:
                    46:0c:3a:fb:64:03:f8:dc:1c:86:e0:36:81:91:01:
                    f2:71
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Netscape Comment:
                Generated by the Security Server for z/OS (RACF)
            X509v3 Subject Alternative Name:
                email:bastila@madeup.com, email:keyleth@madeup.com, email:bobgens@madeup.com,
DNS:ww2.madeup.com, DNS:ww3.madeup.com, DNS:www.madeup.com,
URI:http://www.madeup.com/main.html, URI:ldap://www.madeup.com/ldap:user=client, IP
Address:9.117.24.161, IP Address:9.117.24.162, IP
Address:2001:DB8:3333:4444:5555:6666:7777:8888, IP Address:9.117.24.160
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment, Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Subject Key Identifier:
                7B:7A:3E:87:10:C5:43:11:1F:53:06:78:F3:B1:F1:06:48:06:75:B5
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        47:17:e1:99:0c:e0:e4:c6:43:60:4d:fc:05:b4:f0:85:6e:c0:
        a8:18:a1:a2:d9:73:8b:2a:4f:ff:57:76:21:46:4f:d2:bc:d8:
        ef:cb:24:5a:14:45:d4:61:b6:37:64:82:27:62:7a:0d:56:8b:
        69:3e:46:ce:66:ce:b4:22:b1:c5:2e:0a:cd:f9:7a:13:e5:7f:
        e5:43:78:b7:dd:a1:fa:81:22:66:80:7d:05:8b:81:84:37:2c:
        3b:e4:46:0d:b1:69:82:4c:4b:9f:7d:9a:8f:81:1d:d7:14:ab:
        9c:d8:4a:0b:d6:62:ef:e5:bc:85:16:2d:8f:11:2f:cd:83:db:
        52:78:1a:82:2e:29:45:36:30:2a:ea:9e:09:77:15:41:db:29:
        2f:15:d0:3d:a5:09:9a:ce:05:67:ea:97:4c:82:c7:f9:e9:4c:
        96:67:0a:7b:bc:5c:c3:98:36:de:60:42:41:26:18:2b:99:f0:
        ae:74:19:e3:9f:ec:51:d0:da:33:ab:a0:b8:94:fc:21:19:bc:
        54:37:65:4c:2a:09:4a:0e:6d:d1:a9:28:05:90:45:1e:00:7f:
        e0:c1:48:23:0a:f8:c9:21:b5:54:70:0b:b6:0e:6b:e2:af:2e:
        3d:ae:68:f3:18:43:24:29:8e:0f:f8:27:53:42:97:20:88:d0:
        a2:dc:a7:74
-----BEGIN CERTIFICATE-----
<Omitted for brevity>
-----END CERTIFICATE-----
```

RACDCERT GENCERT Multiple SANs

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

SMF Unload IRRADU00 report for a RACDCERT GENCERT multiple SAN certificate

The first 1452 columns of the output for the single SMF entry is displayed to the left – there is additional information in further columns.

The RACD_SPECIFIED field, which displays a *likely* RACDCERT command that could be issued to achieve the result achieved by the *actual* command, begins in column 1024.

Note the abbreviation scheme used to reduce the amount of space consumed by the SAN values in RACD_SPECIFIED.

```
1                                                                                                                   132
|-------------------------------------------------------------------------------------------------------------------|
RACDCERT SUCCESS  09:55:36 2024-10-24 IM13 NO   NO   NO   IBMUSER  SYS1     NO   YES  NO   NO   NO   NO   NO   NO   NO   NO   YES  N
133                                                                                                                 264
|-------------------------------------------------------------------------------------------------------------------|
O   NO   NO   NO   NO   000 NO   NO   LOCALF10 IBMUSER  08:49:23 2024-10-24          NO   NO   NO   NO   NO   NO   NO   NO   NO   NO
265                                                                                                                 396
|-------------------------------------------------------------------------------------------------------------------|
   SYSMULTI 77F0                     NO   YES  NO   NO   NO   YES  NO   NO   NO   NO   TSO      NO   NO   NO   SYSMULTI
397                                                                                                                 528
|-------------------------------------------------------------------------------------------------------------------|
                        LOCALF10 TERMINAL IBMUSER  SYS1     YES  YES  00
529                                                                                                                 660
|-------------------------------------------------------------------------------------------------------------------|


661                                                                                                                 792
|-------------------------------------------------------------------------------------------------------------------|
                                                    CN=BobGens1
793                                                                                                                 924
|-------------------------------------------------------------------------------------------------------------------|


925                                                                                                                1056
|-------------------------------------------------------------------------------------------------------------------|
                                                                                                  CERTAUTH GENCERT SUBJECTSDN(CN('
1057                                                                                                               1188
|-------------------------------------------------------------------------------------------------------------------|
BobGens1')) SIZE(2048) NOTBEFORE(DATE(2024/10/24) TIME(00:00:00)) NOTAFTER(DATE(2025/10/24) TIME(23:59:59)) KEYUSAGE(HANDSHAKE) ALTN
1189                                                                                                               1320
|-------------------------------------------------------------------------------------------------------------------|
AME(AIP(9.117.24.161 ...(00003)) ADOMAIN('ww2.madeup.com' ...(00002))  AEMAIL('bastila@madeup.com' ...(00002)) AURI('http://www.made
1321                                                                                                               1452
|-------------------------------------------------------------------------------------------------------------------|
up.com/main.html' ...(00001))) WITHLABEL('BOBGENS1')
```

RACDCERT LIST Multiple SANs

RACDCERT LIST (and LISTCHAIN and CHECKCERT): Now lists all SANs in the certificate.

```
Subject's Name:
        >CN=samplecert.O=Test.SP=Poughkeepsie.C=US<
Subject's AltNames:
    IP: 127.0.0.5
    IP: 127.0.0.6
    IP: 127.0.0.7
    EMail: admin1 at us.ibm.com
    EMail: admin2 at us.ibm.com
    Domain: developer.ibm.com
    Domain: demo.ibm.com
    Domain: api.ibm.com
    Domain: tester.ibm.com
    URI: https://developer.ibm.com/welcome.html
    URI: https://tester.ibm.com/token
```

RACDCERT LIST Auth Key ID and Sub Key ID

RACDCERT LIST (and LISTCHAIN and CHECKCERT): Now lists the Authority Key ID and Subject Key ID (that already exists in the certificate).

- **Authority Key ID** – Hash of issuer's (CA's) public key
- **Subject Key ID** – Hash of subject's (this certificate's) public key

```
Start Date: 2023/02/01 00:00:00
End Date: 2024/02/01 23:59:59
Serial Number:
>05<
Issuer's Name:
>CN=sampleCA.O=Test.SP=Poughkeepsie.C=US<
Authority Key ID:
FC:C4:91:DF:D3:32:F3:08:9B:85:42:E9:46:53:C6:61:
97:FE:94:4E
Subject's Name:
>CN=samplecert.O=Test.SP=Poughkeepsie.C=US<
Subject's AltNames:
... ... ...
Subject Key ID:
D8:38:7A:E5:58:3E:79:74:83:66:53:C6:61:97:04:DA:
DC:98:96:2B
... ... ...
```

GRANULAR DATASET ENCRYPTION (RACF)

Statements of Direction

- **Encryption of tape data sets**

  IBM intends to **enhance pervasive encryption** to perform **encryption within the access methods for tape data sets**. It is expected to be **transparent to the application** program **unless it uses EXCP**. This new data set encryption support is intended to be independent of any encryption that occurs in the tape subsystem.

Granular Data Set Encryption

- **A new field in the DFP segment of the DATASET profile specifies encryption policy for *tape, PDSE, and sequential basic and large format* data sets covered by the profile**
  - *No change to current support of extended format data sets*
- **This policy is not bound to the encryption key label in the DFP DATAKEY field**
  - *That is, the key can be sourced elsewhere, like today*
- **For each type, you can choose to**
  - ***IN**clude the type for encryption*
  - ***EX**clude the type from encryption*
  - *Defer to SMS for the decision (the default). SMS checks a FACILITY profiles for system-wide default policy.*

How do I turn it on?

ADDSD and ALTDSD Commands

```
[ DFP(
      [RESOWNER(userid or group-name) | NORESOWNER]
      [DATAKEY(CKDS key label) | NODATAKEY]
      [ENCRYPTTYPES(
                [ALL |
                [INTAPE | EXTAPE | NOTAPE]
                [INPDSE | EXPDSE | NOPDSE]
                [INSEQ  | EXSEQ  | NOSEQ ]
                ]
                    ) | NOENCRYPTTYPES]
    ) | NODFP ]
```

- **ALL is mutually exclusive with EXxxxx and NOxxxx**
- **NOxxxx, INxxxx, and EXxxxx are mutually exclusive for the same type**

```
IRR52128I Mutually exclusive operands are specified for keyword
ENCRYPTTYPES. Processing terminated.
```

Examples

LISTDSD – Examples

```
INFORMATION FOR DATASET BRUCE.* (G)

DFP INFORMATION
---------------
RESOWNER= NONE
DATAKEY= MYKEY
DATA SET TYPES ENCRYPTED= INTAPE EXSEQ

INFORMATION FOR DATASET BRUCE.* (G)

DFP INFORMATION
---------------
RESOWNER= NONE
DATAKEY= MYKEY
DATA SET TYPES ENCRYPTED= ALL INTAPE INPDSE INSEQ
```

USER QUARANTINE

Statement of Direction

Anomaly Detection, Notification, Quarantine:

IBM® plans to provide a software solution that introduces **cyber anomaly detection and notification** for the z/OS® platform to mitigate the potential risk of malicious software. IBM plans to provide **the option of quarantine functionality** that further extends existing remediation options. It is the intent for these combined functions, per NIST guidelines, to be used by the client to **satisfy compliance regulations** requiring anti-malware coverage for z/OS. This intent includes standards such as the Payment Card Industry Data Security Standard (PCI DSS) version 4.0.

10 September 2024 Announcement: IBM Threat Detection for z/OS 1.1 delivers AI-driven discovery of anomalies that could be indicative of a cyberattack

Link: https://www.ibm.com/docs/en/announcements/threat-detection-zos-11-delivers-ai-driven-discovery-anomalies-that-could-be-indicative-cyberattack

Background

User Authentication:

- Authorized applications call RACROUTE REQUEST=VERIFY
  - Validates the user provided credentials and creates an ACEE

Resource Authorization:

- Admin and/or resource owner can create profiles in RACF
  - Define access control security policy for RACF protected resources
  - Profiles can be read into memory by RACLIST for high performance

- Resource managers can call RACROUTE REQUEST=AUTH
  - Asks RACF if a user has the requested authority to access resource

User Quarantine

REVOKE Attribute:

- Prevents a user from successfully authenticating to z/OS applications
- Revoking a user with an **active** session does not always remove their ability to continue to use RACF protected resources from the application

- **Exceptions:**
  - RACF does issue an ENF signal when a user is revoked, and some applications listen for the ENF signal and subsequently prevent the user from performing further actions.
  - Some applications have a way for a user's active session to be cancelled. Like the console command for TSO: C U=USERID

User Quarantine Objective:

- Provide a way for a RACF administrator to prevent a user with an **active** security context from continuing to access RACF protected resources.

User Quarantine - ALTUSER

Quarantining a User:

- Starting in z/OS 3.2, the **ALTUSER** command can be used to contain a user ID, which revokes the user ID and denies access to any RACF protected resources, even for currently active security contexts.

Syntax:

```
ALTUSER userid

…
[ REVOKE [(date)] | NOREVOKE |
  CONTAIN | NOCONTAIN |
  NEVERCONTAIN | ALLOWCONTAIN ]
```

User Quarantine – ALTUSER Keywords

CONTAIN

Specifies that RACF is to prevent the user from accessing the system and immediately fail the user's subsequent access requests, even for currently active sessions.
If a RESUME date exists for this user, it is removed if CONTAIN is specified for the user.
This option also sets the REVOKE attribute in the user profile.

NOCONTAIN

Specifies that RACF is to allow access requests for the user in an active session to function normally.
The NOCONTAIN operand removes the user from the user containment list and removes the CONTAIN attribute from the user profile. NOCONTAIN has no effect on the REVOKE setting. To remove the REVOKE setting for a user, you must specify RESUME.

NEVERCONTAIN

Specifies that the user cannot be contained.
If the user is contained when this attribute is set, the user remains contained. In this case, it is necessary to enter a separate ALTUSER command with the NOCONTAIN option to remove the user from containment.

ALLOWCONTAIN

Removes the NEVERCONTAIN attribute from the user.

RESUME

Keyword is unchanged, but will fail if CONTAIN is already set for user. In this case, NOCONTAIN must be specified on the same command or on a previous command to process the RESUME keyword.

Note: *All containment keywords are mutually exclusive with the REVOKE keyword.*

User Quarantine - ADDUSER

Deleting a Contained User:

- Does not remove it from the containment list, so that any active sessions for that user do not gain the ability to access RACF protected resources.

ADDUSER NOCONTAIN:

- The ADDUSER command has the NOCONTAIN keyword to be able to add back a user that is currently contained.

Syntax:

```
ADDUSER userid
...
   [NOCONTAIN]
```

NOCONTAIN:

If specified for a user ID that exists in the user containment list, the NOCONTAIN operand causes the user ID to be removed from the containment list, and thus removed from quarantine.

Usage note: The NOCONTAIN keyword is needed when the user ID that is being defined through ADDUSER was deleted after previously being placed in the containment list. NOCONTAIN removes the ID from that list and allows it to be defined again as a user that is able to access the system, based on the usual defined authorities.

User Quarantine – Authorization

SHARE
EDUCATE • NETWORK • INFLUENCE

User Quarantine Authorization:

- The ability to perform User Quarantine can be authorized by RACF SPECIAL or access to profiles in the FACILITY class.

Details:

- **SPECIAL** Authority required to use containment keywords - OR -
- **FACILITY** class profile **IRR.CONTAIN.USER:**
  - **READ** – Allows use of CONTAIN keyword (ALTUSER or ADDUSER)
  - **UPDATE** – Allows use of CONTAIN, NOCONTAIN, NEVERCONTAIN and ALLOWCONTAIN keywords

User Quarantine – LISTUSER

LISTUSER for a Contained User:

- The LISTUSER command will now show CONTAIN status as a user attribute.

Note: This example shows a user that was CONTAINed prior to NEVERCONTAIN being set

```
USER=ERIC   NAME=UNKNOWN   OWNER=IBMUSER    CREATED=23.237
DEFAULT-GROUP=SYS1      PASSDATE=N/A     PASS-INTERVAL=N/A
PHRASEDATE=N/A
ATTRIBUTES=REVOKED CONTAINED NEVERCONTAIN GRPACC ATTRIBUTES=PROTECTED
 REVOKE DATE=NONE    RESUME DATE=NONE
 LAST-ACCESS=UNKNOWN
 CLASS AUTHORIZATIONS=NONE
 NO-INSTALLATION-DATA
 NO-MODEL-NAME
...
```

User Quarantine – SETROPTS

Active Containment list:

- The active containment List has the users CONTAINed **since last IPL**.
- Previously CONTAINed users are REVOKEd so cannot initiate sessions, so they do not need to be in the active containment list.

SETROPTS Lists the Active Containment List:

- SETROPTS lists the users in the active user containment list.
- Does not list all users with the CONTAIN attribute since that would require reading all user profiles.

SETROPTS LIST Examples:

- **When No users contained:**

```
CONTAINED USERS: THERE ARE NO CONTAINED USERS
```

- **When some users contained:**

```
CONTAINED USERS:
  FRED     RACFUSR1 RACFUSR2
```

Apar Information

RACF User Quarantine support has been shipped back to z/OS 3.2 and 3.1:

- Base support shipped via APARs:
  - OA67286 (RACF 3.1 & 3.2)
  - OA67288 (SAF 3.1)
- z/OS 2.5 has coexistence support for a shared DB that maintains containment settings when FLAG4 is updated
  - OA67786 (RACF 2.5)

RACDCERT IDENTITY TOKENS

Identity Token Support

Identity Token:

- An Identity Token is used to encode details about a user 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:

- Support is added to RACROUTE authentication processing to generate and validate Identity Tokens (IDT).
- **Generation -** An application can call RACROUTE or initACEE and request that an IDT be returned.
- **Validation -** An application can call RACROUTE to authenticate a user with an IDT specified instead a credential like a password.

IDT Configuration:

- The security administrator can create profiles in the IDTDATA class to configure how certain fields in an IDT are generated and validated.
- The profiles can be used to control options such as which key is used to sign the IDT and its validity period.

Identity Token – Use Cases (1 of 3)

SHARE
EDUCATE • NETWORK • INFLUENCE

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.

IDT – Use Cases (2 of 3)

SHARE
EDUCATE • NETWORK • INFLUENCE

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

IDT – Use Cases (3 of 3)

SHARE
EDUCATE • NETWORK • INFLUENCE

Generate IDT from existing ACEE security environment:

- In some cases, an application needs to flow work onto another LPAR for a protected user.

Problem:

- Protected users can not be authenticated with password/PassTicket/JWT.

Solution:

- Applications can generate an IDT for an existing ACEE security environment (for a protected or non-protected user ID) using the SAF initACEE callable service.
- Protected users can be authenticated with RACROUTE REQUEST=VERIFY with a JWT generated by initACEE.

IDT - New in 3.2 and OA65299

Starting with OA65299 (RACF) and OA66783 (SAF) for z/OS 3.1:

- Add support to generate and validate RACF IDTs with RSA signatures using secure ICSF CCA key labels.
- Add support to generate and validate RACF IDTs with HMAC signatures using clear and secure ICSF CCA key labels.

Value:

- More secure and streamlined key distribution for RSA keys via digital certificates.
- Option for CCA HMAC keys provides more flexibility and opportunity to use CCA secure HMAC keys in installations that do not have an crypto co-processor in EP11 mode.

IDT Configuration

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.

IDT – New IDTPARMS Keywords

IDTPARMS segment RALTER command keywords

[IDTPARMS(

[SIGTOKEN(*pkcs11-token-name*) | NOSIGTOKEN]

[SIGSEQNUM(*pkcs11-sequence-number*) | NOSIGSEQ]

[SIGCAT(pkcs11-category) | NOSIGCAT]

[SIGLABELPRIMARY(*primary-label*)]

[SIGKIDPRIMARY(*primary-kid*)]

[SIGALG(HS256 | HS384 | HS512 |

RS245 | RS384 | RS512) | NOSIGALG]

[ANYAPPL(YES | NO)]

[IDTTIMEOUT(*timeout-minutes*)]

[PROTALLOWED (YES | NO)]

)

NOIDTPARMS]

Location of the signing key (PKCS#11 TKDS HMAC key)
Note: Does not support RSA

Location of the signing key (CCA CKDS RSA or HMAC key)

Key Identifier (KID) of SIGLABELPRIMARY

Signature algorithm to use

Whether IDTs can be used by other applications

Validity interval of a token

Whether IDT can authenticate a protected ID

IDT – KID – Key Identifier

Using the new SIGKIDPRIMARY keyword, installations can configure a KID value to be logically associated with the key identified by SIGLABELPRIMARY.

- The consumer of the IDT can use the KID claim to help identify the appropriate key for signature validation.

KID Claim use in IDT Generation:

- When an IDT is signed with the SIGLABELPRIMARY, the value identified by SIGKIDPRIMARY is set in the KID claim in the IDT.

KID Claim use in IDT Validation:

- When an IDT contains a KID claim and the IDTDATA class profile contains a SIGKIDPRIMARY:
  - These values must match before the key identified by SIGLABELPRIMARY is used to validate the signature.
  - When they do not match, signature validation is not attempted, and a failure return code is returned

Note: The SIGKIDPRIMARY value is not used when an IDT is signed by or validated with a key from the SIGTOKEN keyword.

IDT Configuration

RACDCERT to provision RSA Key Pair:

- **Generate a key pair in the PKDS with RACDCERT:**
  ```
  RACDCERT GENCERT WITHLABEL('TEST.RSA2048') RSA(PKDS(*)) SIZE(2048)
  SUB(CN('TEST.RSA2048'))
  ```
  - **Note:** RACDCERT ADD with PKDS keyword can be used to add an existing certificate and keys to the PKDS.
- **Define a covering IDTDATA profile:**
  ```
  RDEF IDTDATA JWT.TSOIM13.*.SAF IDTPARMS(SIGLABELPRIM('TEST.RSA2048')
  SIGALG(RS256))
  ```
- **Generated JWT RACROUTE REQUEST=VERIFY for RACFU01:**
  Header: `{"alg": "RS256"}`
  Body:
  ```
  {"jti":"00E04D2A6355877296000000B500001","txn":"00E04D2A6355877296000000B
  500001","iss":"saf","sub":"RACFU01","aud":["TSOIM13","*ANYAPPL*"],"iat":173
  6963652,"exp":1736963952,"amr":["saf-pwd"]}
  ```
  Signature: Binary RSA signature

Apar Information

RACF IDT support for RSA and HMAC CCA keys has been shipped back to z/OS 3.1:

- OA65299 (RACF 3.1 & 3.2)
- OA66783 (SAF 3.1)

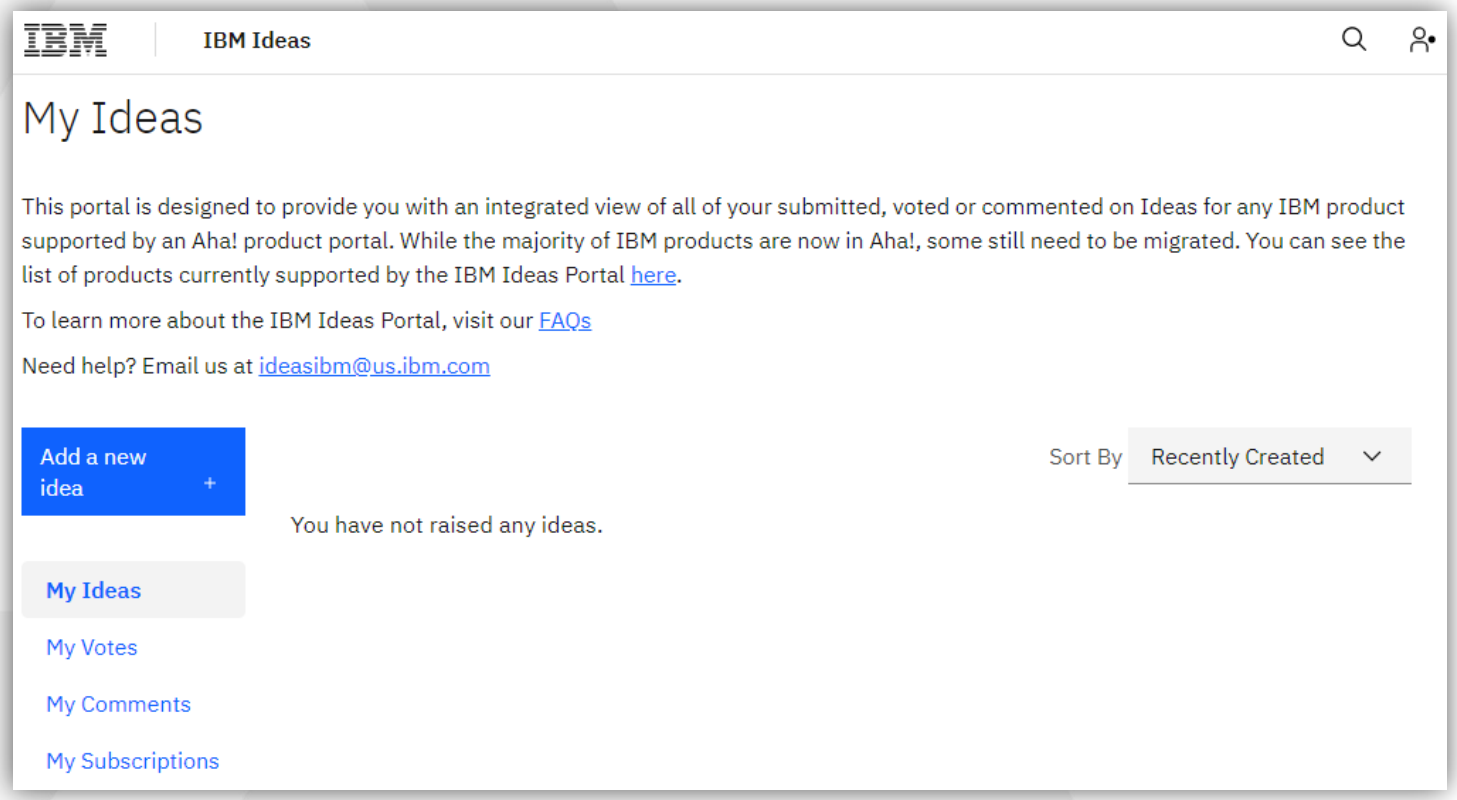
IDEAS

IBM Ideas

Requirements may be submitted to IBM Ideas:

- Reviewed by the design and development teams
- Facilitates a dialog between clients and IBM

Ideas Link: https://ideas.ibm.com



Experience more with IBM

Visit us at the IBM Booth #113

After a full day of technical sessions, take a break with us!

Connect with our experts, snap a photo with the z17 Plexi or the latest Telum II, and get an up-close look at our Spyre Accelerator.

Come back each day for fresh topics and demos at our expert stations.

Think 2026

Join 5000+ senior business and technology leaders who are seizing the AI revolution to unlock unprecedented growth and productivity at Think 2026.

Find out more information using the QR code below.

IBM Digital Asset Haven

IBM Digital Asset Haven is the operational backbone for financial institutions and regulated enterprises entering the digital asset economy.

Find out more information using the QR code below.

Your feedback is important!

SHARE
EDUCATE • NETWORK • INFLUENCE

Submit a session evaluation for each session you attend:

www.share.org/evaluation

BACKUP