z/OS 2.4 IBM Education Assistant (IEA)

Solution (Epic) Name: MFA Enhancements – Identity Token Support

Element(s)/Component(s): RACF

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

- Overview
- Identity Tokens (IDT) and JSON Web Tokens (JWT)
- IDT Generation and Validation via RACROUTE REQ=VERIFY
- IDT Configuration via new IDTDATA class and IDTPARMS segment
- Auditing

Identity Token Support Overview

- **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 authentication processing 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 it’s validity period.

Identity Token Support Overview …

Linking Multiple Authentication API Calls:

- In some cases, an application which performs user authentication may be required to call the RACROUTE authentication APIs multiple times to complete the authentication process. Applications can use an IDT to allow RACROUTE to link authentication status information between these multiple authentication API calls.
  - **Authentication Scenarios with multiple API calls:**
    - Some authentication scenarios require multiple API calls to authenticate a user:
      - **Expired Password / Invalid New Password / MFA Expired PIN …**
    - When an one time use MFA credential is used, it is consumed on the first API call, and the subsequent calls fail.
  - **TSO will be adding support, to improve end-user MFA logon experience**

Replaying Proof of Authentication:

- Some applications have a need to authenticate a user and then replay that authentication multiple times. The Identity Token support allows applications to authenticate a user and receive proof of that authentication. The returned IDT can be specified on subsequent calls to RACROUTE instead of other authentication credentials like a password.
  - **Caching Authentication:**
    - 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.
    - Signed JWTs can be returned to an end user for later use by the application.

Identity Token Format – JSON Web Tokens (JWT)

Identity Token support is designed to be flexible and support different types of Identity Tokens going forward.

- Initially, RACROUTE will have support for an IDT in the format of a JSON Web Token (JWT).

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

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

JWT Properties:

- **Pronounced** – JWT is pronounced "jot"
- **Compact** - Intended for space constrained environments
- **Signed** – Can be signed to provide integrity of the claims.
  - JWS – JSON Web Signature – Signed JWT
- **Encryption** – Can be encrypted to provide secrecy of claims.
  - JWE – JSON Web Encryption – Encrypted JWT
- **Base64** – Can be encoded in base64url for ease of use in web environments
  - Base64url(JOSE Header).Base64url(JWS Payload).Base64url(JWS Signature)
- **Encoded** – Encoded in UTF8 (Must generate and accept JWTs in UTF8).
  - The Base64url encoded version must also be in UTF8 as that is the source for the signature.

JWT – JSON Example

JWT:

- **Header – (JOSE):**
  - {“alg”:“HS256”}
- **Body Claims – (JWS Payload):**
  - {“jti”:“cb05…”,“iss”:“saf”,“sub”:“USER01”,“aud”:“CICSLP8”,“exp”:1486744112.473}
- **Signature - (JWS):**
  - x’389A21CD32108C3483DA’

JWT base64 Encoded:

- Base64url(JOSE Header).Base64url(JWS Payload).Base64url(JWS Signature)
- 2Ce23xe490fG989N.m239vm3N29U2n2pN9mA02l32lMn.8Nw5JlK2nQnc241Okl

JWT – JSON Claims

- **JWT:**
  - **Header (JOSE):**
    - **{“alg” : “HS256” or “none”}** – Signature Algorithm: **HS256** = HMAC with **SHA-256**, none = unsecured (Other algorithms supported)
  - **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

RACF and IBM MFA IDT Usage

- **Generation of IDT/JWTs:**
  - **RACF & SAF RACROUTE REQ=VERIFY is responsible for creating the IDT/JWT:**
    - “amr” ”mfa” claim values are based on the MFA PC Return Codes
    - “amr” “saf” claim values are based on the SAF Password/Phrase validation status
    - RACF will sign the JWTs – Key is selected based on a IDTDATA policy profile
    - Expiration time – Configured in JWT policy profile

- **Validation of IDT/JWTs:**
  - **RACF is responsible for parsing and validating input IDTs/JWTs:**
    - Validate the JWT including the signature and expiration time
  - **IBM MFA is responsible for reacting to the “amr” "mfa" claim in the JWT:**
    - RACF sends IBM MFA the JSON body via the IBM MFA PC
    - IBM MFA returns the same RCs as though it has performed the action stated in a successful “amr” “mfa” claim.
  - **RACINIT is responsible for reacting to the "amr" "saf" claim in the JWT:**
    - RACINIT will return the same RCs as though it has performed the action stated in a successful “amr” “saf” claim.

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.

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 APAR OA55926/OA55927, the RELEASE keyword corresponds to a parameter list version number. Version PLV0001 is the initial parameter list version number and contains all parameters in RELEASE=77B0 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 APAR OA55926

Administrative Control over IDTs

- Security administrators can control the use of tokens by defining profiles in the new IDTDATA general resource class, using a new IDTPARMS segment
  - The IDTDATA class must be ACTIVE before Identity Tokens will be generated or validated by RACF authentication processing.
  - The IDTDATA class must be ACTIVE and RACLISTed before any profiles in the class will be used by RACF authentication processing.
- For any combination of application and user ID, you can specify:
  - The signature algorithm to use
  - The location of the signing key
  - Validity interval of a token
  - Whether the token can be used by applications other than the one requesting its creation

Administrative Control over the use of IDTs …

- IDTDATA profile format is

  <IDT Type>.<application name>.<user ID>.<IDT issuer name>

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

- Generics are allowed. When a user is authenticated with a JWT, RACF looks for the best matching profile from which to retrieve settings

Administrative Control over the use of 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 the use of IDTs …

- Example:

```
RDEFINE IDTDATA JWT.APPL01.USER01.SAF
        IDTPARMS(SIGTOKEN(mytoken) SIGALG(HS256)
                 ANYAPPL TIMEOUT(30))
```

- Applies to authentication of USER01 as requested by the APPL01 application
- Identifies the PKCS#11 (TKDS) token name of ‘mytoken’
  - The installation must create this key in ICSF. There is no command in RACF to generate the key.
- Specifies that the HS256 (HMAC256) signature algorithm is used
- Specifies that the IDT created by ‘APPL01’ may be used to authenticate via any other application
- Specifies that the IDT times out after 30 minutes (the default is 5 minutes)

Enhanced SMF Record

- SMF 80 relocate section 443 was introduced with original MFA support
- It contains information about what authenticator was used
- It is included in event code 1 records (RACROUTE REQUEST=VERIFY)

- A reserved bit will be used to indicate that the format has been extended
- New information will be included to indicate that a user was authenticated with an Identity Token.

RACROUTE EXITs and IDTDATA class active

- The ICHRIX01 preprocessing and ICHRIX02 post processing exits can alter the behavior of RACROUTE REQ=VERIFY authentication processing.
- When the IDTDATA class is activated, RACROUTE REQ=VERIFY will begin processing the IDTA parameter for any application which has specified it.
- The Identity Token represents a new way to authenticate a user with RACROUTE REQ=VERIFY. In many cases when an Identity Token is specified, the password and password phrase parameters will be ignored.
- Before activating the IDTDATA class, the installation must ensure that any ICHRIX01 and ICHRIX02 exits are compatible with Identity Token processing. For example, if these exits inspect the password or password phrase parameters to make processing decisions, they must take into account the new RACROUTE IDTA parameter processing.

Interactions & Dependencies

- To exploit this item, all systems in the Plex must be at the new z/OS level: No

- Software Dependencies
  - ICSF with TKDS for generating and validating signed IDTs
  - IBM MFA

- Hardware Dependencies
  - None

- Exploiters
  - TSO Logon Processing

TSO Exploitation

- 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 like: Expired MFA PIN or expired Password, RSA Next Token Code Mode and MFA protocols which required multiple steps.
  - New Message – “IKJ56469I ENTER MFA INFORMATION”
    - MFA requires user requires more information to complete the authentication process.
- Support is not activated in RACF until the IDTDATA class is ACTIVE.
  - When IDTDATA class is not ACTIVE, the IDTA parameter is ignored by RACF.

Migration & Coexistence Considerations

- Toleration/coexistence APARs/PTFs: None
- Migration actions: None
- Coexistence actions: None

Installation

- List anything that a customer needs to be aware of during install:
  - None

Session Summary

- RACF/SAF authentication processing can generate and validate Identity Tokens (IDT).
- Applications can use Identity Tokens to improve the authentication experience, especially for MFA users.
- The security Administrator can configure how Identity Tokens are generated and validated.

Appendix

- Ross Cooper (rdc@us.ibm.com)

- Publications:
  - z/OS Security Server RACF Command Language Reference
  - z/OS Security Server RACROUTE Macro Reference
  - z/OS Security Server RACF Callable Services
  - z/OS Security Server RACF Macros and Interfaces
  - z/OS Security Server RACF Data Areas
  - z/OS Security Server RACF Messages and Codes