SHARE
EDUCATE • NETWORK • INFLUENCE

JSON Web Tokens and RACF

RACF Identity Token Support

Ross Cooper, CISSP
IBM z/OS Security Server Design and Development
rdc@us.ibm.com

IDENTITY TOKEN SUPPORT

RACF Identity Token Availability

- **Base release of z/OS V2R4**

- **APARs for V2R2 / V2R3:**
  - RACF - OA55926: NEW FUNCTION - IDENTITY TOKEN SUPPORT
  - SAF - OA55927: NEW FUNCTION - IDENTITY TOKEN SUPPORT

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.
- The RACF 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 – Use Cases

Two main use cases for Identity Token Support:

- Replaying Proof of Authentication

- Linking together Multiple Authentication API Calls

Identity Token Support – Use Cases

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.

Identity Token Support – Use Cases

SHARE
EDUCATE • NETWORK • INFLUENCE

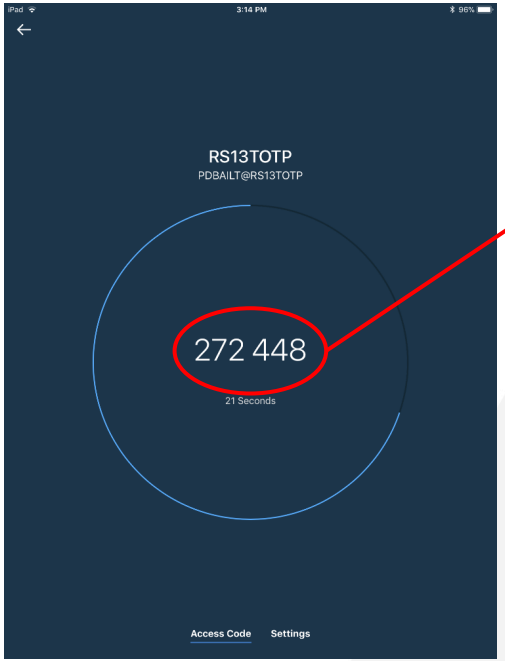
Linking Multiple Authentication API Calls:

- 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 information between multiple authentication API calls without replaying the MFA credentials.

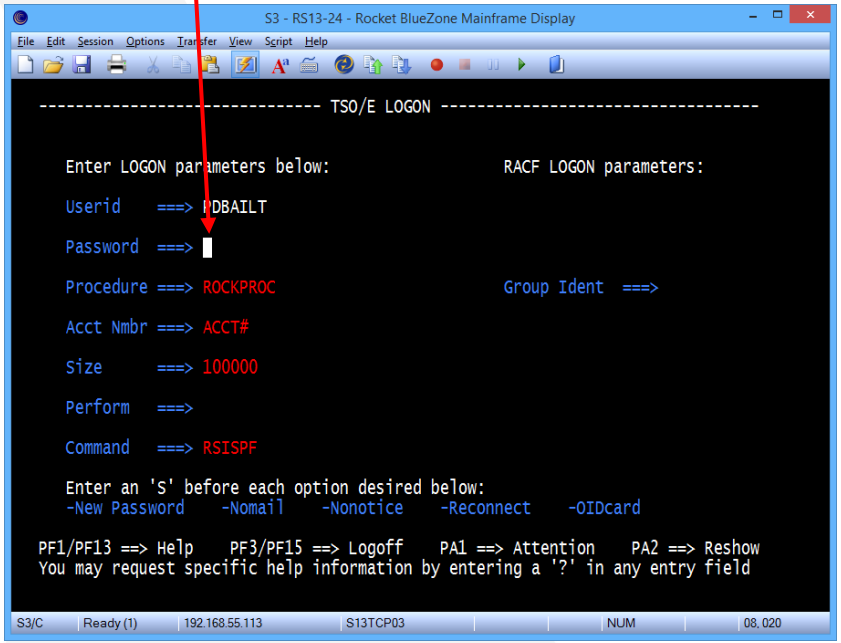
IBM z MFA - Compound In-band



Password: passw0rd

272448

272448:passw0rd



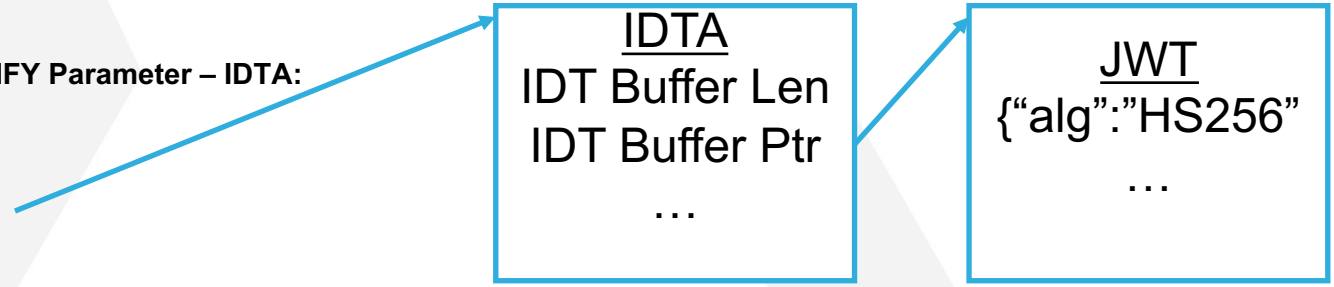
- Compound in-band allows users to authenticate with both a one-time-use token code and RACF password
- What happens if the RACF password expires?
  - Application calls RACF again with the new password
  - Token code was already “consumed”
  - JWT support addresses this issue

Identity Token Externals – RACROUTE REQ=VERIFY

- **Authentication**
  - z/OS Applications authenticate users by gathering credentials and calling RACROUTE REQ=VERIFY
  - RACROUTE REQ=VERIFY callers can use the IDTA parameter to generate and validate JWTs
  - The IDTA points to an area which describes where the JWT is to be returned or specified along with other details.
  - Mapped by SAF Macro: **IRRPIDTA**

- **New RACROUTE REQ=VERIFY 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.

JWT – JSON Web Token

SHARE
EDUCATE • NETWORK • INFLUENCE

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

Signed and Unsigned JWTs

- RACROUTE can generate and validate signed or unsigned IDTs.
- Profiles in the IDTDATA class can be created to configure when IDTs are signed.

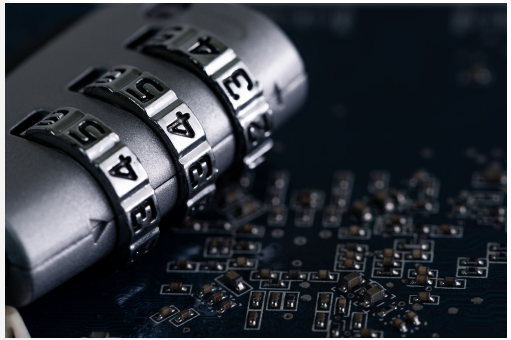
- **Unsigned IDTs:**
  - An authorized application which keeps an unsigned IDT under its own control and does not accept an IDT from an end user may generate and specify an unsigned IDT.
  - RACROUTE will not accept an unsigned IDT from an end user.
  - The TSO IDT support does not require signed IDTs (not accepted from end users).

- **Signed IDTs:**
  - Signed IDTs can be returned to end users and accepted from end users.

- **End Users:**
  - When an Application indicates the IDT will be generated for an “end user”:
    - Signed Identity Tokens must be configured to be returned
  - When an Application indicates the IDT has been proved from an “end user”:
    - RACROUTE will require the provided IDT to be signed.

Protecting a JWT

- When an application receives an Identity Token (IDT) from RACROUTE, it must keep it in a protected location.
  - Must be kept as protected as a password
  - An IDT can be used to authenticate a user though RACROUTE and therefore must be kept in protected storage.
- Some applications may allow IDTs to be specified from an end user.



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)]

[IDTTIMEOUT(*timeout-minutes*)]

)

NOIDTPARMS]

Location of the signing key

Signature algorithm to use

Whether IDTs can be used by other applications

Validity interval of a token

ICSF Key Details

- **The ICSF PKCS#11 Token Handle is comprised of three parts:**
  - **SIGTOKEN(*pkcs11-token-name*):** 32 Bytes – A-Z (Case insensitive) + @,#,$ and period symbol
  - **SIGSEQNUM(*pkcs11-sequence-number*):** 8 byte – Hex
  - **SIGCAT(*pkcs11-category*):** ‘T’ for clear key or ‘Y’ for secure key

- **PKCS#11 Handle format is described in the ICSF Publication:**
  - z/OS ICSF Application Programmer's Guide
    - Chapter 4. Introducing PKCS #11 and using PKCS #11 callable services

- **Creating the Signing key:**
  - There is no RACF interface to create the IDT signing key. The installation must create the key in the TKDS with an ICSF Callable Service:
    - PKCS#11 Generate Secret Key (CSFPGSK) (or)
    - PKCS#11 Token Record Create (CSFPTRC)

Identity Token Signature Algorithm

- **Identity Token Signature algorithm to use**
  - IDT may be signed or unsigned
    - Not all application usage will require a signed JWT (TSO does not)
  - RACF supports HMAC signature algorithm

- **SIGALG(HS256 | HS384 | HS512) | NOSIGALG**

  - **HS256**
    - HMAC with SHA-256 (Default)
  - **HS384**
    - HMAC with SHA-384
  - **HS512**
    - HMAC with SHA-512

ANYAPPL – IDT can be used for any application name?

- **ANYAPPL(YES | NO):**
  - Specifies whether the IDT that RACROUTE generates can be used for any application name or only for the application name that performed authentication. The default value is YES.

  - When ANYAPPL(YES) is specified, RACROUTE will generate the IDT so it can be used for any application name.
    - "*ANYAPPL*" - Is included in the audience claim value

  - When ANYAPPL(NO) is specified, RACROUTE will generate the IDT so that it can only be used by the application name that performed authentication.

IDTTIMEOUT – IDT Validity Period

- **IDTTIMEOUT(*timeout-minutes*):**
  - Specifies the number of minutes that the Identity Token (IDT) associated with the profile is active.
  - RACROUTE uses this value to calculate the expiration date for the IDT.
  - The value of timeout-minutes can be between 1 and 1440.
  - The default value is 5.

- **Expired IDTs are not accepted by RACROUTE:**
  - RACROUTE RC: 8/6C/F - IDT Expiration Date indicates IDT is expired.

Identity Token Configuration Examples

SHARE
EDUCATE • NETWORK • INFLUENCE

Example RDEFINE:

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

Example RLIST:

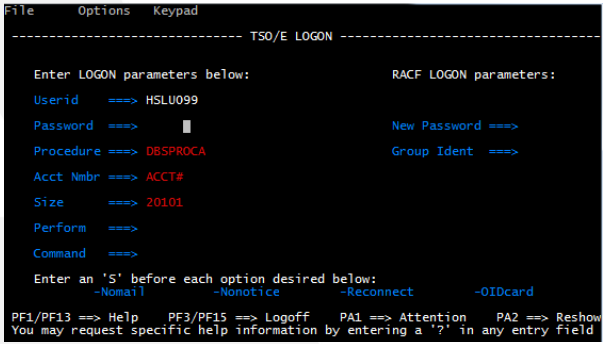
```
RLIST IDTDATA JWT.APPL01.USER01.SAF IDTPARMS
...
IDTPARMS INFORMATION
--------------------
SIGNATURE TOKEN NAME = MYTOKEN
SIGNATURE SEQUENCE NUMBER = 00000001
SIGNATURE CATEGORY = T
SIGNATURE ALGORITHM = HS256
IDT TIMEOUT = 00000030
ANYAPPL = YES
```

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 Z 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.



CICS Support for Identity Tokens

- **CICS provides support for JSON Web Tokens (JWTs):**
  - CICS applications can convert basic authentication credentials of a user to a JWT and then validate users with the JWT.

- **Avoids replay of one-time-use MFA credentials:**
  - This is particularly useful where applications currently using passwords are being converted to using MFA tokens.

- **CICS VERIFY TOKEN:**
  - Using the **VERIFY TOKEN** command, CICS can convert basic authentication credentials of a user to a JWT and then validate the JWT.
    - User is authenticated with BASICAUTH:
      - Password/Phrase/MFA credentials
    - VERIFY TOKEN can return a JWT in OUTTOKEN

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

RACF JWT Support – Current Limitations

- **JWT Validation:**
  - Only supports local z/OS RACF user registry
  - Can not pass in a JWT from another user registry and have z/OS map the external identity to the local z/OS user ID.
  - In the future:
    - RACF could use z/OS RACF defined mappings to determine RACF user ID
    - RACF could use JWT claim specified RACF user ID with registry identifier

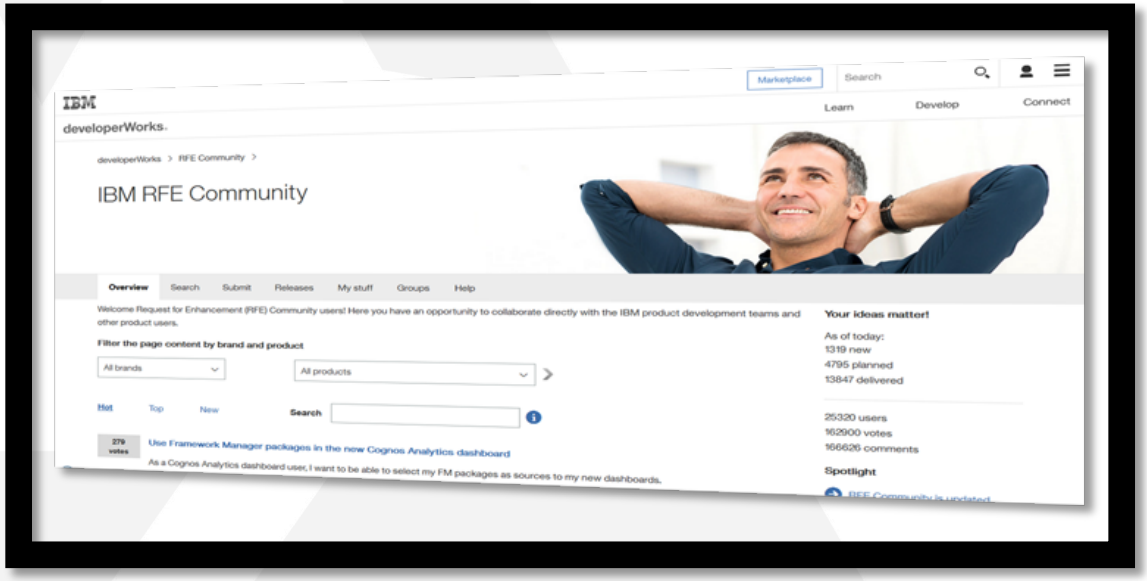
- **JWT Generation:**
  - Included JWT claims are not configurable.
  - Can not customize a RACF created JWT for consumption by another user registry Identity Provider

- **JWT Signature:**
  - Only supports HMAC
  - Does not support RSA – Certificate based asymmetric crypto signatures

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 Trademarks

SHARE
EDUCATE • NETWORK • INFLUENCE

Trademarks:

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