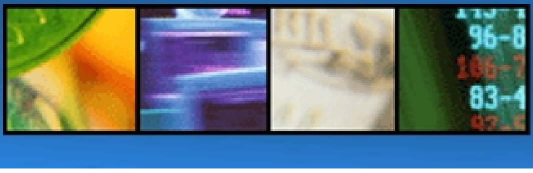
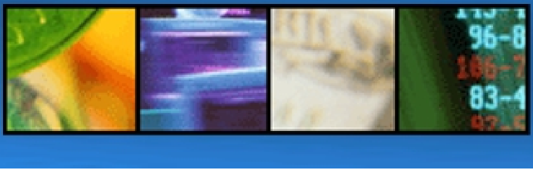


Extended DB2 DRDA Security

A DB2 for z/OS subsystem which receives connect requests from other DB2 for z/OS subsystems via TCP/IP must pass the USERID credentials to SAF for authentication. Usually, for TCP/IP communications, the USERID is presented with a password for authentication. Unlike SNA communications, specific TCP/IP requesting locations cannot be configured to be trusted to provide already-verified USERIDs. TCP/IP communications must therefore be setup at a requesting DB2 for z/OS to send both a USERID and a password in the connect request to a remote DB2 location.

Table of content:

This is an excerpt of common IBM DB2™, z/OS™, RACF™ and other product documentation. Important aspects have been picked out and emphasized for better representation of causal relationships between underlaying technologies.



Managing TCP/IP-based connection requests

DRDA connections that use TCP/IP have fewer security controls than do connections that use SNA protocols. When planning to control inbound TCP/IP connection requests, you must decide whether you want the requests to have authentication information, such as RACF passwords, RACF PassTickets, and Kerberos tickets, passed along with authorization IDs.

> ➔ To protect your authentication information, use the z/OS Communications Server IP Application Transparent Transport Layer Security (AT-TLS) to secure your network connections. To complement the use of AT-TLS, set the TCPALVER subsystem parameter of installation panel DSNTIP5 to SERVER_ENCRYPT. Setting this parameter to SERVER_ENCRYPT provides the strongest level of security. Connections are accepted only if user credentials are provided to authenticate the user ID, and strong encryption is used to protect the user ID and credentials.

How to manage inbound TCP/IP-based connection requests:

- For requests that use RACF passwords or PassTickets, enter the following RACF command to indicate which user IDs that use TCP/IP are authorized to access DDF (the distributed data facility address space):

```
PERMIT ssnm.DIST CLASS(DSNR) ID(yyy) ACCESS(READ)
  WHEN(APPCPORT(TCPIP))
```

  <u>Consider the following questions:</u>

  Do you permit access by TCP/IP?
  If the serving DB2 for z/OS subsystem has a DRDA port and resynchronization port specified in the BSDS, DB2 is enabled for TCP/IP connections.

  Do you manage inbound IDs through DB2 or RACF?
  All IDs must be passed to RACF or Kerberos for processing. No option exists to handle incoming IDs through DB2.

  Do you trust the partner?
  TCP/IP does not verify partner LUs, as SNA does. If your requesters support mutual authentication, use Kerberos to handle this authentication on the requester side.

  If you use passwords, are they encrypted?
  Passwords can be encrypted through:

  - RACF using PassTickets
  - DRDA password encryption support. DB2 for z/OS as a server supports DRDA-encrypted passwords and encrypted user IDs with encrypted passwords. See Security mechanisms for DRDA and SNA for more information about using DRDA encryption.

  If you use Kerberos, are users authenticated?
  If your distributed environment uses Kerberos to manage users and perform user

authentication, DB2 for z/OS can use Kerberos security services to authenticate remote users.

Do you translate inbound IDs?
Inbound IDs are not translated when you use TCP/IP.

How do you associate inbound IDs with secondary IDs?
To associate an inbound ID with secondary IDs, modify the default connection exit routine (DSN3@ATH). TCP/IP requests do not use the sign-on exit routine.

To receive requests from a DB2 for z/OS requester over TCP/IP connections that use RACF-protected user IDs and RACF PassTickets (as passwords), you must take the following additional actions in RACF:

1. Create a RACF PTKTDATA resource profile at the server system or sysplex by issuing one of the following commands:
   ```
   RDEFINE PTKTDATA IRRPTAUTH.applname.useridRDEFINE
   PTKTDATA IRRPTAUTH.applname.*
   ```

   where …
   ■ *applname* is either the generic LU name, the IPNAME assigned to each member of a serving data sharing group, or the LUNAME or IPNAME assigned to the serving non-data sharing subsystem.
   ■ *userid* is either an asterisk ("*") or a RACF-protected user ID that you want to allow into the serving subsystem or a member of a data sharing group.

2. Refresh and load the PTKTDATA resource profile by issuing the following command:
   ```
   SETROPTS RACLIST(PTKTDATA) REFRESH
   ```

3. Permit the user ID that is assigned in the STARTED profile in the ssidDIST address space to read the new profile by issuing one of the following commands:
   ```
   PERMIT IRRPTAUTH.applanme.userid CLASS(PTKTDATA) -
       ID(dist_userid) ACCESS(READ)  PERMIT
   IRRPTAUTH.applname.* CLASS(PTKTDATA) -
       ID(dist_userid) ACCESS(READ)
   ```

   where *userid* and *dist_userid* are not the same.

Prepare requestors for access requests using extended security

The communications database (CDB) is a set of DB2 catalog tables that let you control aspects of remote requests. Columns in the SYSIBM.LUNAMES, SYSIBM.IPNAMES, SYSIBM.USERNAMES, and SYSIBM.LOCATIONS tables pertain to security that related to the requesting system.

IPNAMES

The SYSIBM.IPNAMES table is used only for outbound requests that use TCP/IP protocols.

LINKNAME CHAR(8)

The name used in the LINKNAME column of SYSIBM.LOCATIONS to identify the remote system.

IPADDR

Specifies an IP address or domain name of a remote TCP/IP host.

SECURITY_OUT

Indicates the DRDA® security option that is used when local DB2® SQL applications connect to any remote server that is associated with this TCP/IP host.

A

The letter A signifies the security option of already verified, and it is the default. Outbound connection requests contain an authorization ID and no password. The value that is used for an outbound request is either the DB2 user's authorization ID or a translated ID, depending on the value in the USERNAMES column.

The authorization ID is not encrypted when it is sent to the partner. For encryption, see option D.

R

The letter R signifies the RACF PassTicket security option. Outbound connection requests contain a user ID and a RACF PassTicket. When a RACF PassTicket is generated, the LINKNAME column value is used as the RACF PassTicket application name and must match the following at the target server

- LUNAME - if the remote site is a DB2 subsystem that is defined with only an LUNAME value and no GENERIC LU name value or IPNAME value
- GENERIC - if the remote site is a DB2 subsystem that is defined with a GENERIC LU name value in addition to an LUNAME value but no IPNAME value
- IPNAME - if the remote site is a DB2 subsystem that is defined with an IPNAME value that triggers the remote DB2 subsystem's DDF to activate only its TCP/IP communications support.

The value that is used for an outbound request is either the DB2 user's authorization ID or a translated ID, depending on the value in the USERNAMES column. The translated ID is used to build the RACF PassTicket. Do not specify R for CONNECT statements with a USER parameter.

The authorization ID is not encrypted when it is sent to the partner.

D

The letter D signifies the security option of user ID and security-sensitive data encryption. Outbound connection requests contain an authorization ID and no password. The authorization ID that is used for an outbound request is either the DB2 user's authorization ID or a translated ID, depending on the USERNAMES column.

This option indicates that the user ID and the security-sensitive data are to be encrypted. If you do not require encryption, see option A.

E

The letter E signifies the security option of user ID, password, and security-sensitive data encryption. Outbound connection requests contain an authorization ID and a password. The password is obtained from the SYSIBM.USERNAMES table. The USERNAMES column must specify "O".
This option indicates that the user ID, password, and security-sensitive data are to be encrypted. If you do not require security-sensitive data encryption, see option P.

P
The letter P signifies the password security option. Outbound connection requests contain an authorization ID and a password. The password is obtained from the SYSIBM.USERNAMES table. If you specify P, the USERNAMES column must specify "O".
If you specify P and the server supports encryption, the user ID and the password are encrypted. If the server does not support encryption, the user ID and the password are sent to the partner in clear text. If you also need to encrypt security-sensitive data, see option E.

USERNAMES CHAR(1)
This column indicates whether an outbound request translates the authorization ID. When you specify O, use the SYSIBM.USERNAMES table to perform the translation.
O
The letter O signifies an outbound ID that is subject to translation. Rows in the SYSIBM.USERNAMES table are used to perform ID translation. If a connection to any remote server is to be established as trusted, a row in the SYSIBM.USERNAMES table is used to obtain the system authorization ID.
S
The letter S signifies the system authorization ID, within a trusted context, obtained from the SYSIBM.USERNAMES table. If the system authorization ID that is specified in the AUTHID column is different from the primary authorization ID, DB2 sends the user switch request on behalf of the primary authorization ID after successfully establishing the trusted connection.
blank
No translation is done.

LOCATION

The SYSIBM.LOCATIONS table contains a row for every accessible remote server. Each row associates a LOCATION name with the TCP/IP or SNA network attributes for the remote server. Requesters are not defined in the SYSIBM.LOCATIONS table.

LOCATION CHAR(16)
Indicates the unique location name by which the the remote server is known to local DB2® SQL applications.
LINKNAME CHAR(8)
Identifies the VTAM or TCP/IP network locations that are associated with this row. A blank value in this column indicates that this name translation rule applies to any TCP/IP or SNA partner.

If you specify a nonblank value for this column, one or both of the following situations must be true:

- A row exists in table SYSIBM.LUNAMES that has an LUNAME value that matches the LINKNAME value that appears in this column.
- A row exists in table SYSIBM.IPNAMES that has a LINKNAME value that matches the LINKNAME value that appears in this column.

PORT CHAR(32)

Indicates that TCP/IP is used for outbound connections when the following statement is true:

- A row exists in SYSIBM.IPNAMES, where the LINKNAME column matches the value that is specified in the SYSIBM.LOCATIONS LINKNAME column.

If the previously mentioned row is found, and the SECURE column has a value of 'N', the value of the PORT column is interpreted as follows:

- If PORT is blank, the default DRDA port (446) is used.
- If PORT is nonblank, the value that is specified for PORT can take one of two forms:
    - If the value in PORT is left-justified with one to five numeric characters, the value is assumed to be the TCP/IP port number of the remote database server.
    - Any other value is assumed to be a TCP/IP service name, which you can convert to a TCP/IP port number by using the TCP/IP getservbyname socket call. TCP/IP service names are not case-sensitive.

If the previously mentioned row is found, and the SECURE column has a value of 'Y', the value of the PORT column is interpreted as follows:

- If PORT is blank, the default secure DRDA port (448) is used.
- If PORT is nonblank, the value that is specified for PORT takes the value of the configured secure DRDA port at the remote server.

TPN VARCHAR(64)

Used only when the local DB2 begins an SNA conversation with another server. When used, TPN indicates the SNA LU 6.2 transaction program name (TPN) that will allocate the conversation. A length of zero for the column indicates the default TPN. For DRDA conversations, this is the DRDA default, which is X'07F6C4C2'.

DBALIAS(128)

Used to access a remote database server. If DBALIAS is blank, the location name is used to access the remote database server. This column does not change the name of any database objects sent to the remote site that contains the location qualifier.

TRUSTED

Indicates whether the connection to the remote server can be trusted. This is restricted to TCP/IP only. This column is ignored for connections that use SNA.

Y = The location is trusted. Access to the remote location requires a trusted context that is defined at the remote location.

N = The location is not trusted.

SECURE

Indicates the use of the Secure Socket Layer (SSL) protocol for outbound connections when local DB2 applications connect to the remote database server by using TCP/IP.

Y = A secure SSL connection is required for the outbound connection.

N = A secure connection is not required for the outbound connection.

USERNAMES

The USERNAMES table contains information that is needed for outbound translation only. It has the following columns:

TYPE CHAR(1)
Whether the row is for outbound translation. The value 'O' is valid for TCP/IP connections.
AUTHID CHAR(8)
Authorization ID to translate. If blank, it applies to all authorization IDs.
LINKNAME CHAR(8)
Identifies the TCP/IP network location associated with the row. A blank indicates it applies to all TCP/IP partners. For nonblank values, this value must match the LINKNAME value in SYSIBM.IPNAMES.
NEWAUTHID CHAR(8)
The translated value of AUTHID.
PASSWORD CHAR(8)
The password to accompany an outbound request. This column is ignored if RACF® PassTickets, or already verified USERIDs are used.

Remember: Inbound ID translation and come from checking are not done for TCP/IP requesters.

Setting DB2 for z/OS DRDA AR to use encrypted passwords

If you are connecting from DB2 for z/OS (as DRDA AR, application requestor), you need to populate the valid user ID and password in the communication database (CDB) tables to connect to remote DB2 for z/OS server. If you are using INSERT statements to populate the CDB tables, you need to set your user IDs and password in clear text. If the network is connected to the Internet or a wide-range of intranets this can be a security exposure.

DB2 for z/OS provides the DSNLEUSR stored procedures to let users store encrypted translated authorization ID (NEWAUTHID) and password (PASSWORD) in the SYSIBM.USERNAMES table.

To use the DSNLEUSR stored procedures, z/OS Integrated Cryptographic Service Facility (ICSF) must be installed, configured, and active.

Note: From an administration and security point of view, you should restrict access to the DSNLEUSR stored procedure to security or database administrators.

There is no way to tell the value of encrypted translated authorization IDs or encrypted passwords once they are inserted in CDB table. If you need to update a password, you need to DELETE the current row before executing the DSNLEUSR stored procedure again.

Using ICSF on z/OS

In the z/OS environment, the Integrated Cryptographic Service Facility (ICSF) provides access to cryptographic functions through callable services.

You can use z/OS Security Server RACF to control which applications can use specific keys and services, which can help you ensure that keys and services are used only by authorized users and jobs. You can also use RACF to audit the use of keys and services. The XCSFKEY class controls who can export a token using the Symmetric Key Export callable service (CSNDSYX). To set up these controls, you create and maintain RACF general resource profiles in the CSFKEYS class, the CSFSERV class, and the XFACILIT class. The CSFKEYS class controls access to cryptographic keys with the key label, the CSFSERV class controls access to ICSF services, and resources in the XFACILIT class define a key store policy that controls the use of key tokens that are stored in the CKDS and PKDS.

DB2 for z/OS requires to have ICSF on z/OS to use AES encryption. Data stream encryption also requires ICSF on z/OS. If ICSF is not enabled, you will see the message in SYSLOG:
DSNL046I -D9C3 DSNLTSEC ICSF NOT ENABLED

Message: DSNL046I   csect-name ICSF is not enabled

Explanation: This message indicates that a cryptographic service is required, but the Integrated Cryptographic Service Facility (ICSF) is unavailable.

System action: DB2 cannot continue with the encryption or decryption function.

System programmer response: If the cryptographic service facility is not installed, install it before requesting encryption functions. If it is configured, verify that the ICSF service is available and working correctly.

Related reference: ICSF System Programmer's Guide

Using PassTickets

To minimize the storing of USERIDs and passwords in the requesting DB2 for z/OS Communications Data Base (CDB), many users configure the requesting DB2 for z/OS to use RACF PassTickets when connecting to other remote DB2 for z/OS locations.

Since the serving DB2 does not have any indication that the password could be a RACF PassTicket, the serving DB2 passes the USERID and password to SAF for authentication. For most USERIDs, SAF (RACF) will check to see if the password is a valid RACF PassTicket, and if it is, will then authenticate the USERID for access to the serving DB2 for z/OS.

However, if the USERID is a RACF protected USERID, RACF will reject this authentication request by flagging the password as invalid. Both ICH408I and IRR013I messages will be issued indicating that an invalid password was presented for authentication.

How RACF Processes the Password or PassTicket

To validate a password or PassTicket, RACF does the following:

1. Determines whether the value in the password field is the RACF password for the user ID.
    - If it is the RACF password, the validation is complete.
    - If it is not the RACF password, processing continues.
2. Determines whether a secured signon application profile has been defined for the application in the PTKTDATA class.
    - If a profile has not been defined, the user receives a message from the application3 indicating that the password is not valid.
    - If the application is defined in the PTKTDATA class, processing continues.
3. Evaluates the value entered in the password field. The evaluation determines whether:
    - The value is a PassTicket consistent with this user ID, application, and time range.
    - It has been used previously on this computer system for this user ID, application, and time range.
    - If the value was used before, and if PassTicket replay protection (see below) has not been bypassed, the user receives a message from the application4 indicating that the password is not valid.
    - If the value was not used before, the PassTicket is considered valid and processing continues.

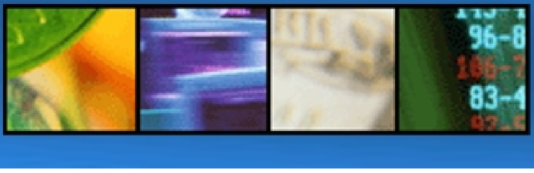
Determines whether the value is a valid PassTicket:

- If the PassTicket is valid, RACF gives the user access to the desired application.
- If the value is not valid, the host application sends a message to the user indicating that the password is not valid.

Time Considerations:
A PassTicket is considered to be within the valid time range when the time of generation, with respect to the clock on the generating computer, is within plus or minus 10 minutes of the time of evaluation, with respect to the clock on the evaluating computer.
Be sure that your MVS system and the evaluating computer use clock values that are within that time range. RACF uses the value stored for coordinated universal time (UTC), formerly called Greenwich mean time (GMT), in the algorithms that process PassTickets.
One way to ensure that reasonably synchronized values are used is to set UTC in the GMT value of the MVS time of day (TOD) clock and to set a similar value in each of the other systems with which RACF shares PassTicket information. You can still use the MVS local time for local timestamp information, and resetting the local time does not affect the GMT value kept in the TOD clock.

Bypassing PassTicket Replay Protection
You might use the option to bypass PassTicket replay protection when the threat of PassTicket replay is not a security concern, such as in the following cases:

- Multiple end-users who share the same user ID
- Trusted registry domains that exchange PassTickets as a method of establishing trust
- Applications that request PassTickets for a particular USERID/APPLID combination more than once during a one-second time interval.



The option to bypass PassTicket replay protection allows the plus-or-minus-10-minute PassTicket replay protection to be bypassed for selected applications or combinations of selected applications, users, or groups.

You indicate that replay protection is to be bypassed for a particular application by adding the text string NO REPLAY PROTECTION to the APPLDATA field of the PTKTDATA profile for that application. You must separate each word in the string with a single blank space, alphanumeric character, or keyboard symbol. The NO REPLAY PROTECTION text string will always be translated to upper case by the RALTER or RDEFINE commands.

The NO REPLAY PROTECTION text string can appear anywhere within the APPLDATA field, allowing for the existence of other information already in the field, or for new information that might be added in the future.

The following are examples of commands that will cause PassTicket replay protection to be bypassed:

```
RALTER  PTKTDATA profile-name APPLDATA('NO REPLAY PROTECTION')
RDEFINE PTKTDATA profile-name APPLDATA('NO REPLAY PROTECTION')
RDEFINE PTKTDATA profile-name
   APPLDATA('FOR THIS APPLICATION NO REPLAY PROTECTION IS IN
EFFECT')
```

The role of TCPALVER system parameter

The TCPALVER subsystem parameter specifies whether DB2 is to accept TCP/IP connection requests that contain only a user ID (no password, RACF PassTicket, or Kerberos ticket). Or, this parameter specifies if a stronger form of security is required. This parameter is not relevant to trusted context users that have been switched.

YES|CLIENT	A new connection is accepted with a user ID only.
è Security credentials such as a password are not required to authenticate the user ID that is associated with the connection.

NO|SERVER	A user ID and password are required for connection requests, or the connection must be authenticated by a RACF PassTicket or Kerberos ticket. The user ID and password can be encrypted or non-encrypted.

SERVER_ENCRYPT	A user ID and password are required for connection requests. Kerberos tickets are also accepted. In addition, one of the following must be true:

- The user ID and password is AES (Advanced Encryption Standard)-encrypted.
- The connection is accepted on a port that ensures AT-TLS (Application Transparent - Transport Layer Security) policy protection, such as a DB2 security port (SECPORT).

Non-encrypted security credentials or RACF PassTickets are not accepted unless the connection is secured by the TCP/IP network. RACF PassTickets are encoded, which is considered to be a form of

security that is weaker than encryption. DES (Data Encryption Standard)-based encryption is also considered insecure.
This option provides the best security. Connections are accepted only if user credentials are provided to authenticate the user ID, and strong encryption is used to protect the user ID and credentials

Troubles using PassTickets

Unable to obtain a PassTicket - 00D31059

An abend occurred, reason code 00D31059 issued. The attempt to access the remote database resource failed, and the failure is reported to the application.

An attempt to allocate a conversation to the remote site failed because DB2 was unable to obtain a RACF PassTicket. The user specified an 'R' in the SECURITY_OUT column of the SYSIBM.IPNAMES or SYSIBM.LUNAMES communications database (CDB) tables for the partner site. As a result, DB2 invokes RACF to extract a PassTicket for the partner site. However, RACF could not provide a PassTicket, and the attempt failed.

The error usually occurs due to incorrect or missing RACF definitions. To avoid this error, specify the proper RACF definitions to provide for the PassTicket. Alternatively, you may avoid the use of PassTickets by changing the SECURITY_OUT column of the SYSIBM.IPNAMES and/or SYSIBM.LUNAMES CDB table for the partner site.

Invalid PassTicket passed - 00F30085

An abend occurred, reason code 00F30085 issued. The attempt to access the remote database resource failed, and the failure is reported to the application.

Potential causes for 00F30085 (requester system provided password could not be verified) authentication failures related to use of DB2 z/OS and RACF Pass Tickets:

- Users intermittently receive 00F30085.
  The issue is that a RACF PASSTICKET has a TIME BASED ALGORITHM. Based on this algorithm, duplicate PassTickets will be generated within a 1 second interval. This means that if multiple authentications are processed within a 1 second interval, the authentication will fail at the DB2 z/OS server with a 00F30085 reason code.
  The work around provided by RACF support to bypass this condition is to use no replay with RACF 'NO REPLAY PROTECTION'.

- A 00F30085 condition can occur when the "key" (appname) used to generate the PassTicket at the DB2 z/OS requester system is different than the "key" used at the DB2 z/OS server to parse the PassTicket.

  If a DB2 z/OS server subsystem IS a member of a data sharing group, then a Generic LU name must be defined, and also be common to all members of the group, and thus
  the Generic LU name will effectively be given to RACF for PassTicket parsing purposes.

As a result, any remote DB2 z/OS requester subsystems accessing the remote group should use the group's Generic LU name as the LINKNAME column value of the SYSIBM.LOCATIONS row associated with the remote group. If the DB2 z/OS requester SYSIBM.LOCATIONS LINKNAME column value is not identical to the DB2 z/OS server's Generic LU name, then PassTicket authentication will fail.

If a DB2 z/OS server subsystem IS NOT a member of a data sharing group, then the Generic LU name must be null and thus the server subsystem's LU name will effectively be given to RACF for PassTicket parsing purposes.

As a result, any remote DB2 z/OS requester subsystems accessing the remote server subsystem should use the server subsystem's LU name as the LINKNAME column value of the SYSIBM.LOCATIONS row associated with the remote subsystem. If the DB2 z/OS requester SYSIBM.LOCATIONS LINKNAME column value is not identical to the DB2 z/OS server's LU name, then PassTicket authentication will fail.

- The RACF PTKTDATA profile has not been defined or the profile name does not match the Generic LU or LU name (see above).

To avoid 00F30085 within your application you should avoid generating PassTickets within a seconds interval. This could be achieved as follows:

```
-- Series of SQL statement on remote DB server:
-- Connect and authenticate once, then do you work:
CONNECT TO REMTSRV1;
SELECT CAST(CURRENT SERVER AS CHAR(8)) AS SERVER,
       CAST(CURRENT SQLID AS CHAR(8)) AS SQLID
FROM SYSIBM.SYSDUMMY1;
SELECT * FROM SYSIBM.SYSTABLES;
SELECT * FROM MYTABLE;
```

Protected Users and PassTickets

Until DB2 Version 9 for z/OS without having installed an special APAR (see below), “protected USERID’s” cannot be used with PassTicket. A very prominent example for an protected user is the TWS (OPC) STC user.

Basically, PassTickets take the place of passwords, and since a protected id cannot have a password, it cannot have a PassTicket either. An alternative would be to make use of so-called surrogate USERID’s. (See next section.)

For IDs who need to have access to remote DB2, turn off the protected attribute or install following APAR.

With APAR PM43292 ("PM43292: Allow RACF protected userids to be PassTicket authenticated", 2011-11-02) DB2 has been changed to support receiving RACF PassTickets with RACF protected USERID's over TCP/IP communications from requesting DB2 for z/OS subsystems.

When receiving RACF PassTickets as passwords with RACF protected USERID's over TCP/IP communications from a DB2 for z/OS requester, the following RACF actions must be taken as follows:

- A RACF PTKTDATA resource profile must be created at the server system or sysplex using the following naming rules:

  ```
  RDEFINE PTKTDATA IRRPTAUTH.applname.USERID or
  RDEFINE PTKTDATA IRRPTAUTH.applname.*
  ```

  Where applname is either the generic LU name or IPNAME assigned to each member of a serving data sharing group or is the LUNAME or IPNAME assigned to the serving non-data sharing subsystem.

  Where USERID is either an asterisk ("*") or a RACF protected USERID that one wants to allow into the serving subsystem or member of a data sharing group.

- Once the RACF profile has been defined, the PTKTDATA resource must be refreshed as follows:

  ```
  SETROPTS RACLIST(PTKTDATA) REFRESH
  ```

- Once the PTKTDATA resource profiles have been refreshed and loaded, the USERID assigned in the STDATA of the STARTED profile of the ssidDIST address space must be permitted to read this new profile as follows:

  ```
  PERMIT IRRPTAUTH.applanme.USERID CLASS(PTKTDATA) -
      ID(dist_USERID) ACCESS(READ) or
  PERMIT IRRPTAUTH.applname.* CLASS(PTKTDATA) -
      ID(dist_USERID) ACCESS(READ)
  ```

  Where USERID and dist_USERID are not the same.

The above actions do not need to be taken if one does not use RACF protected USERID's in connect requests from a requesting DB2 for z/OS to a serving DB2 for z/OS. The RACF resource profile can be created prior to installing the PTF of this APAR. However, until all members of a data sharing group have been started with the PTF applied, some members may still reject the connection attempt as receiving an invalid password when a RACF protected USERID is used in the connection attempt.

Surrogate User Jobs Submission

If you cannot make use of PassTickets with you RACF protected users, you need to allow a user to submit a job on behalf of another user, you can set up surrogate job submission. Profiles in the SURROGAT resource class specify that a user (the surrogate user) is able to submit a job on behalf of another user (the execution user). The surrogate user does not need to supply the execution user's password, but must have read access to the security label under which the job runs. The job runs with the user ID that the jobcard specifies, not the surrogate user's user ID. The audit record for surrogate job submission identifies both the surrogate user and the jobcard user ID.

To define which jobs are allowed to be submitted by surrogate users, the security administrator creates a profile for each appropriate job in the SURROGAT resource class, and permits the submitting user to the access list in the specific job profile with at least READ access. For surrogates of TWS (OPC) user’s the password should never expire. A TSO segment is not necessary – so no one could use that surrogate to login into TSO. The surrogate USERID needs to be connected to RACF group of the protected USERID.

You now could use that surrogate with your batch jobs: Specify the surrogate USERID as parameter USER=id in a JES JOB card without specifying a password.
Example:

```
//REMOTE JOB ‘REMOTE DB2 ACCESS’,CLASS=A,MSGCLASS=B,
//       USER=SURROID1
```

The surrogate user must specify the execution user's user ID on the USER parameter on the JOB statement and must not specify a password. If the PASSWORD parameter is specified with a password, surrogate processing is not performed, and audit records generated by the job's activities do not indicate that the job is a surrogate job. This applies not only to jobs submitted through the TSO SUBMIT command, but any time the surrogate user is a RACF-defined user.

To allow surrogate users, do the following:

1. Ensure that the installation exit for the TSO SUBMIT command (IKJEFF10) does not prevent users from submitting jobs with job names that do not match their user IDs. The installation exit supplied by IBM® meets this requirement, because it does not check the JCL of submitted jobs. For more information, see z/OS TSO/E Customization.
2. If your installation implemented the sample ICHRTX00 exit from SYS1.SAMPLIB member RACINSTL to enable surrogate user processing, you should migrate to profiles in the SURROGAT class. After RACF is installed, the code in the ICHRTX00 exit that checks $SUBMIT.userid profiles is not used. You should copy the $SUBMIT.userid profiles to SURROGAT profiles as follows:
   ```
   RDEFINE SURROGAT execution-userid.SUBMIT
   FROM($SUBMIT.execution-userid) FCLASS(FACILITY)
   ```
3. Define resource profiles in the SURROGAT class for each execution user who needs to allow others to be surrogate users:
   ```
   RDEFINE SURROGAT execution-userid.SUBMIT UACC(NONE)
   OWNER(execution-userid)
   ```

Note: Specifying the OWNER operand allows the execution user to issue the PERMIT command for this profile.

4. To specify that another user can act as the surrogate for an execution user, give the surrogate user READ access authority:

   ```
   PERMIT execution-userid.SUBMIT CLASS(SURROGAT)
   ID(surrogate-userid) ACCESS(READ)
   ```

   Only users and groups that have READ access authority are allowed to submit jobs on behalf of another user.

   To check whether a user can submit jobs for another user, make sure the user (or a group the user is a member of) is in the access list with READ access authority:

   ```
   RLIST SURROGAT execution-userid.SUBMIT AUTHUSER
   ```

5. When you are ready to control access using the SURROGAT profiles, activate the SURROGAT class:

   ```
   SETROPTS CLASSACT(SURROGAT)
   ```

   To disable surrogate support for a particular user, delete the profile for that user. To disable surrogate support for all users, deactivate the SURROGAT class.