RAA11
Introduction to DB2 for z/OS Security

Gayathiri Chandran
IBM Silicon Valley Laboratory
gchandran@us.ibm.com

© 2009 IBM Corporation

DB2 for z/OS

Acknowledgments and Disclaimers:

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

© Copyright IBM Corporation 2012. All rights reserved.

- U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp

IBM, the IBM logo, ibm.com, DB2, RACF are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.sht

Other company, product, or service names may be trademarks or service marks of others.

Agenda

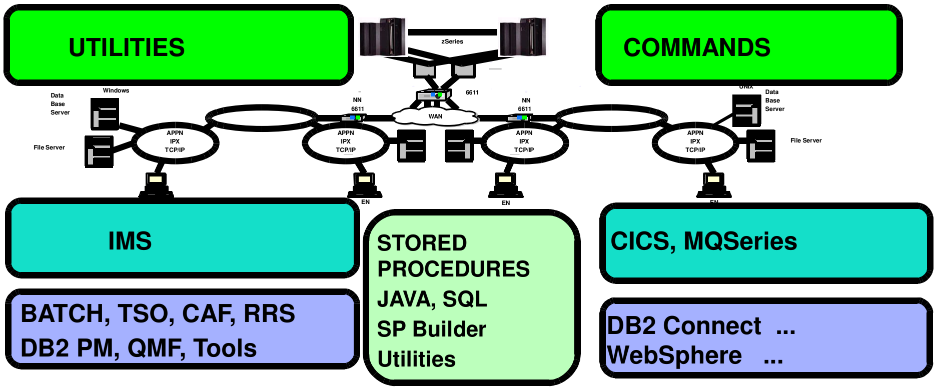
- DB2 security overview
- Controlling access to a DB2 subsystem
- Controlling access to DB2 objects
  - Native DB2 authorization
  - Access Control Authorization Exit authorization
- Security Objects
  - Trusted context and role
  - Row and column access control
- Audit

DB2 Security Overview

- **Security controls access to the DB2 subsystem, its data, and its resources**
  - A security plan sets objectives for a security system
  - Describes how to meet the objectives by using functions of DB2, functions of other programs, and administrative procedures
- **Auditing is how you determine whether the security plan is working and who has accessed data**
  - Auditing includes questions, such as:
    - Have attempts been made to gain unauthorized access?
    - Is the data in the subsystem accurate and consistent?
    - Are system resources used efficiently?

DB2 Operational Environment

- Many ways to access DB2
- DB2 controls all access except the DB2 data sets
- All access to DB2 is authenticated and authorized
  - Authentication process associates a set of IDs with the thread


UTILITIES

COMMANDS

IMS

BATCH, TSO, CAF, RRS
DB2 PM, QMF, Tools

STORED PROCEDURES
JAVA, SQL
SP Builder
Utilities

CICS, MQSeries

DB2 Connect ...
WebSphere ...

Authenticate and authorize a process

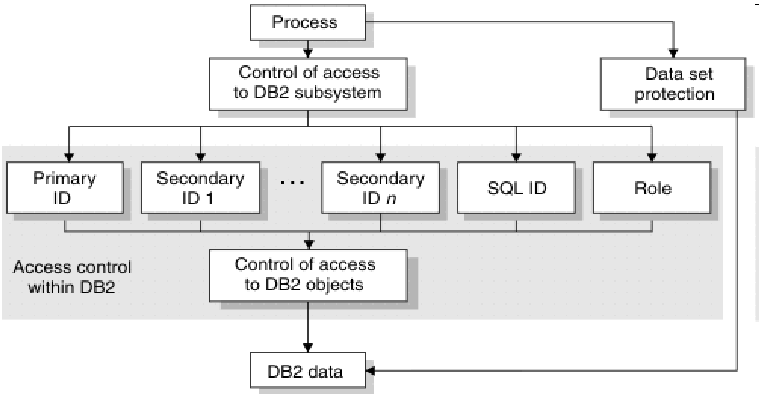
- **Connection processing**
  - Request a new connection to be established
  - Drives connection exit to associate IDs or a Role to a process
  - Request to switch IDs associated with an existing connection when running under a trusted context
- **Sign-on processing**
  - Drives sign-on exit to change the primary ID for an existing connection

Controlling access from an application process

- RACF is used to protect access to a DB2 subsystem
- The RACF resource class used by DB2 is DSNR
- DSNR profiles are created of the form "*subsystem.environment*", where:
  - *subsystem* is the name of a DB2 subsystem
  - *environment* denotes the environment
  - DIST for DDF
  - MASS for IMS (including MPP, BMP, Fast Path, and DL/I batch).
  - SASS for CICS
  - RRSAF
  - BATCH for all others, including TSO, CAF, batch, and all utility jobs
- Security Administrator needs to enable RACF checking for the DSNR class and PERMIT users to access DB2 for a specific environment
- If user ID is authenticated and authorized then process can create a DB2 thread

DB2 Process – Access to data

- **DB2 Process** represents all access to data
- Every process that connects to or signs on to DB2 is represented by one or more DB2 authorization identifiers (IDs)
  - Primary Authorization ID
  - Secondary Authorization IDs (RACF Groups)
  - DB2 role
  - SQL ID



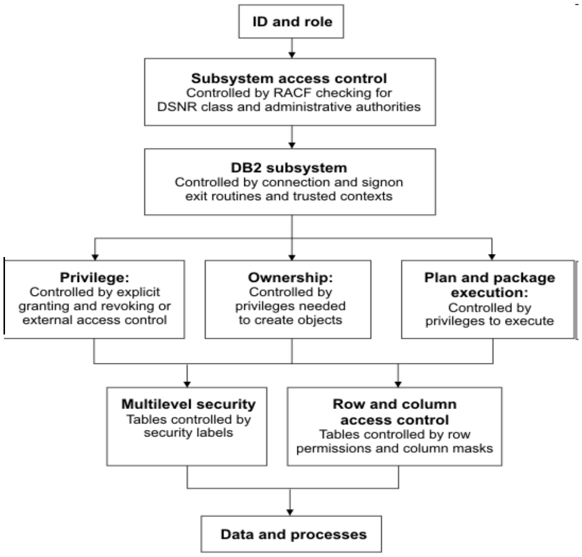
Process
Control of access to DB2 subsystem
Data set protection
Primary ID
Secondary ID 1
. . .
Secondary ID *n*
SQL ID
Role
Access control within DB2
Control of access to DB2 objects
DB2 data

DB2 Process – Authorization IDs

- **RACF User ID**
  - Used by connection and signon exit to generate set of authorization IDs for the thread
- **Primary authorization ID**
  - Generally, identifies a thread. For example, statistics and performance trace records use a primary authorization ID to identify a process
- **Secondary authorization ID**
  - Optional, can hold additional privileges that are available to the process. For example, a secondary authorization ID can be a RACF group ID
- **SQL ID**
  - An SQL ID holds the privileges that are exercised when certain dynamic SQL statements are issued. The SQL ID can be set equal to the primary ID or any of the secondary IDs
- **ROLE**
  - A ROLE is a database entity that groups one or more privileges together and is available only in a trusted connection

Controlling access to DB2 objects

- DB2 controls access to its objects and data by a set of **privileges** through **authorization identifiers (IDs) and roles**
- Each privilege allows a specific action to be taken on an object



ID and role

Subsystem access control
Controlled by RACF checking for DSNR class and administrative authorities

DB2 subsystem
Controlled by connection and signon exit routines and trusted contexts

Privilege:
Controlled by explicit granting and revoking or external access control

Ownership:
Controlled by privileges needed to create objects

Plan and package execution:
Controlled by privileges to execute

Multilevel security
Tables controlled by security labels

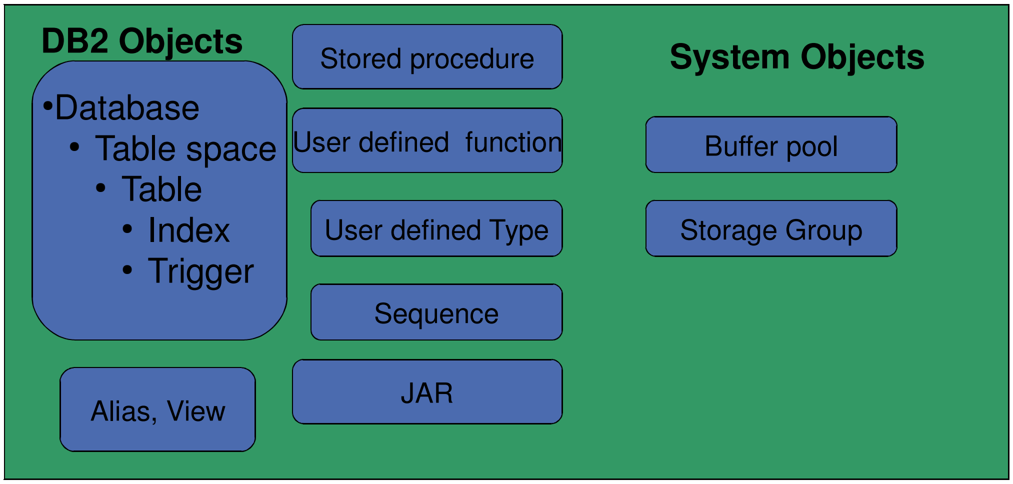
Row and column access control
Tables controlled by row permissions and column masks

Data and processes

Primary ways within DB2 to a thread access to data

DB2 Data Structures

- DB2 catalog is a set of tables which contain information about the data that DB2 is managing
  - All DB2 objects and DB2 managed authorization information


DB2 Objects

- Database
  - Table space
    - Table
      - Index
      - Trigger

Alias, View

Stored procedure

User defined function

User defined Type

Sequence

JAR

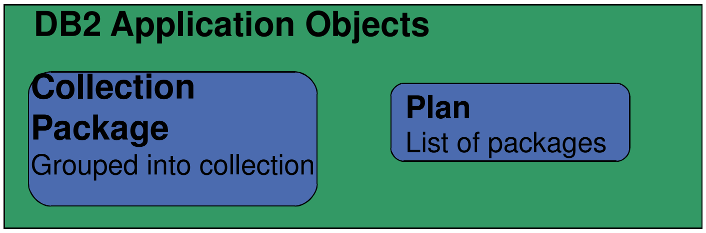
System Objects

Buffer pool

Storage Group

DB2 Application Structures

- Access to DB2 requires an application plan or package
  - Relates an application to an instance of DB2 and processing options
  - Created using DB2 BIND command
- Plan
  - Relates an application process to a local instance of DB2
  - Contains list of packages and specifies processing options
- Package
  - Contains control structures that DB2 uses when it runs SQL statements
  - All control structures in the package are derived from the SQL statements embedded in a single source program


DB2 Application Objects

Collection
Package
Grouped into collection

Plan
List of packages

DB2 Privileges and Authorities

- **Privilege** allows a specific function, sometimes on a specific object
  - Explicit privilege
  - Implicit owner privileges
    - Cannot be revoked
- **Administrative Authority**
  - Set of privileges, often covering a related set of objects.
    - Example: DBADM, PACKADM
  - Includes privileges that are not explicitly granted
    - Example: Ability to execute BACKUP SYSTEM utility is included in the SYSCTRL authority

DB2 Privileges

Database Privileges
CREATETAB
CREATETS
DISPLAYDB
DROP
IMAGCOPY
RECOVERDB
REORG
REPAIR
STARTDB
STATS
STOPDB
LOAD

Collection Privileges
CREATEIN

Table Privileges
ALTER
DELETE
INDEX
INSERT
SELECT
REFERENCES
TRIGGER
UPDATE

Table Space Buffer Pool Storage Group Privileges
USE

System Privileges
ARCHIVE
BINDADD
BINDAGENT
BSDS
CREATEALIAS
CREATEDBA
CREATEDBC
CREATESG
DISPLAY
MONITOR1
MONITOR2
STOPALL
STOSPACE
TRACE
RECOVER
CREATETMTAB
EXPLAIN (V10)
CREATE_SECURE_OBJECT (V10)

Plan Privileges
BIND
EXECUTE

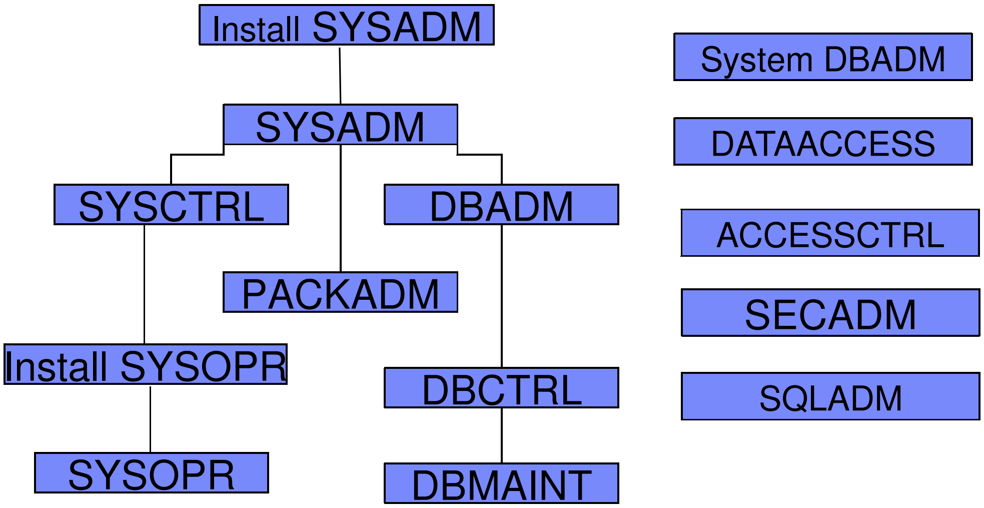
Package Privileges
BIND
COPY
EXECUTE

Schema Privileges
CREATEIN
ALTERIN
DROPIN

Stored Procedure User defined Function Privileges
EXECUTE

JAR User defined Type Privileges
USAGE

DB2 Administrative Authorities



Install SYSADM

SYSADM

SYSCTRL

DBADM

PACKADM

Install SYSOPR

DBCTRL

SYSOPR

DBMAINT

System DBADM

DATAACCESS

ACCESSCTRL

SECADM

SQLADM

DB2 Administrative Authorities

- System authorities specified by installation parameters
  - **Install SYSADM**: Installation authority same as SYSADM
  - **Install SYSOPR**: Installation authority same as SYSOPR
- System administrative authorities
  - **SYSADM**: Includes all DB2 privileges.
  - **SYSCTRL**: Includes all DB2 privileges, except to read or modify user data
  - **SYSOPR**: Allows to issue most commands and execute utilities
- Database authorities
  - **DBADM**: Allows to control and manipulate any table within the database
  - **DBCTRL**: Allows control the database
  - **DBMAINT**: Allows to create objects and run certain utilities on the database
- Application administrative authority
  - **PACKADM**: Includes package privileges on all packages in the specified collection

DB2 10: New System Administrative Authorities

- **SECADM**
  - Performs security related tasks
  - No inherent privilege to access data
- **System DBADM**
  - Allows management of objects in the DB2 subsystem
  - Separates object management from data access and access control
- **DATAACCESS**
  - Access to data in all user tables
  - Execute all plans, packages, functions, procedures
- **ACCESSCTRL**
  - Controls access to data
- **SQLADM**
  - Allows monitoring and tuning without access to data

Implicit privileges through ownership

- DB2 object created by issuing an SQL statement establishes an owner
- The owner of an object implicitly holds all the privileges over that object
  - For example: Tables
    - Alter/drop the table or any index, create index or view, select or update any row or column, insert or delete any row

Privileges exercised through a plan or a package

- DB2 provides a unique access control method for plans and packages to simplify and provide better access control from processes
- The owner of a plan or package can grant the privilege to execute a plan or package to any ID.
- When the EXECUTE privilege on a plan or a package is granted to an ID or ROLE, that ID or ROLE can execute a plan or package without holding the privileges for every action that the plan or package performs
  - Owner of the plan or package is checked for access
- Type of SQL decides when the authorization check is done
  - Static SQL: Authorization checked at BIND or compile time
  - Dynamic SQL: Authorization checked at run time

Example of granting select and execute privileges

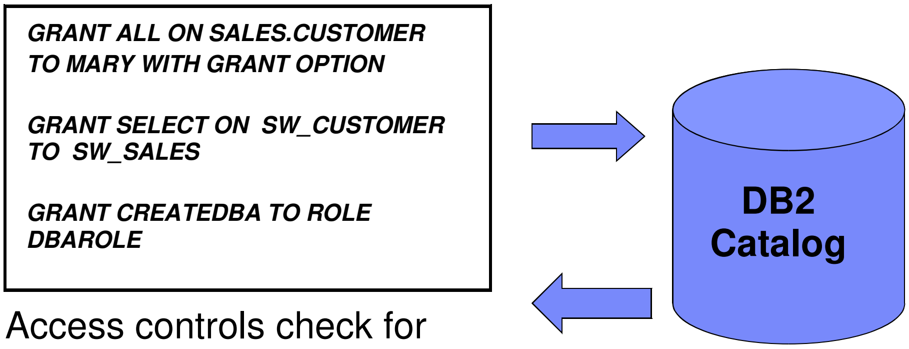
- A program might contain the following statement
  - SELECT * INTO :EMPREC FROM EMPTBL WHERE EMPNO='000010';
- A DBA grants select privilege to the role EMPROLE
  - GRANT SELECT ON TABLE EMPTBL TO ROLE EMPROLE;
- A program with the statement is bound into a package using role EMPROLE as the owner of the package.
  - BIND PACKAGE EMPPKG OWNER(EMPROLE);
- A DBA using role EMPROLE grants execute privilege to EMPUSER
  - GRANT EXECUTE ON PACKAGE EMPPKG TO EMPUSER;
- Any process that executes the packages must have EMPUSER as one of its primary or secondary IDs
- Any process that executes the package is not required to have select privilege on the EMPTBL table

Control of access to DB2 objects

- **DB2 native authorization**
  - Access is controlled by the SQL GRANT and REVOKE statements
  - Stored in DB2 catalog tables
- **Access Control Authorization Exit** (DSNX@XAC)
  - Exit point provided by DB2 which can control access to DB2 resources
  - The programming interface is RACROUTE, which is part of the System Authorization Facility (SAF)
  - Other vendors support the SAF interface

Native DB2 Authorization – SQL GRANT

- SQL GRANT statement grants privileges to authorization IDs, roles
- Delegation via GRANT… WITH GRANT OPTION



```
GRANT ALL ON SALES.CUSTOMER
TO MARY WITH GRANT OPTION

GRANT SELECT ON  SW_CUSTOMER
TO  SW_SALES

GRANT CREATEDBA TO ROLE
DBAROLE
```

Access controls check for granted authority.

DB2 Native Authorization – SQL REVOKE

- SQL REVOKE statement revokes privileges from authorization IDs and roles.
- REVOKE …BY clause allows administrators to revoke privileges granted by others
- Cascading revoke
  - In DB2 10, NOT INCLUDING DEPENDENT PRIVILEGES clause can be specified on the SQL REVOKE statement to avoid cascade revoke
    - System parameter, REVOKE_DEP_PRIVILEGES can be set to control the cascading effect of revoke

```
REVOKE SELECT ON  SALES.SW_CUSTOMER FROM TED NOT
INCLUDING DEPENDENT PRIVILEGES;
```

DB2 - Access Control Authorization Exit

- Exit point is driven
  - Once at DB2 subsystem start up
  - If exit authorization is used:
    - For each DB2 authorization request
    - Once at DB2 subsystem termination
- Exit CSECT name: DSNX@XAC
- Exit parameter list: DSNDXAPL
- DB2 provides dummy DSNX@XAC routine
- DB2 provides sample LKED JCL for DSNX@XAC
  - Install job DSNTIJEX in SDSNSAMP

RACF/DB2 External Security Module

- Fully supported exit module designed to receive control from the DB2 access control authorization exit point
- From DB2 V8, the exit module ships in ‘SYS1.SDSNSAMP(DSNXRXAC)’
- New classes defined in RACF CDT (Class Descriptor Table)
- RACF stores all of its information in the RACF database
- Access to a resource is given using the RACF PERMIT command
- Access to a resource is removed using the RACF PERMIT command with the DELETE keyword

RACF/DB2 External Security Module

- **Initialization**
  - Loads profiles for RACF/DB2 authorization checking
    - Classes targeted for use must be active
  - If unsuccessful or no classes are active, DB2 will not drive the exit point again
- **Authorization Checking**
  - Checks user’s authority to specified DB2 resource
    - Return code 0 – Access allowed
    - Return code 8 – Access not allowed
    - Return code 4 – Don’t know. Defers to DB2 authorization check
- **Termination**
  - Clean up profiles loaded into data spaces

DB2 Objects and their RACF classes

DB2 Object Type	RACF Class Name
Bufferpool	MDSNBP
Collection	MDSNCL
Database	MDSNDB
JAR	MDSNJR
Package	MDSNPK
Plan	MDSNPN
Schema	MDSNSC
Sequence	MDSNSQ

DB2 Object Type	RACF Class Name
Storage group	MDSNSG
Stored procedure	MDSNSP
System	MDSNSM
Table/Index/View	MDSNTB
Table space	MDSNTS
User defined type	MDSNUT
User defined function	MDSNUF

Exploiting RACF multilevel security with DB2

- Row-level security checks allow you to control which users have authorization to view, modify, or perform other actions on specific rows
- Used when mandatory row-level security checks are required
- Multilevel security can be implemented with the following combinations:
  - DB2 authorization and RACF multilevel security
    - DB2 grants are used for authorization at the DB2 object level
    - RACF performs mandatory access checking on DB2 tables using security labels
  - RACF access control and RACF multilevel security
    - RACF is used to control authorization at the DB2 object level and perform mandatory access checking on DB2 tables using security labels

Security objects

DB2 9: Trusted context and Role

- **Trusted context** establishes trust between DB2 and an external entity such as
  - RRSAF (Resource Recovery Services Attachment Facility)
  - DSN Command Processor
  - Application Server
- Once established, a **trusted connection** provides the ability to
  - Efficiently switch user with optional authentication
  - Acquire special set of privileges using a Role
  - Acquire special RACF Security Label authority

Database Role

- Database entity with one or more privileges
- Established only through a trusted connection
- User assigned only one role in a trusted connection
- Can optionally be the OWNER of DB2 objects

```
CREATE ROLE ADMINROLE;

DB2 native authorization – new ROLE keyword for GRANTEE:
    GRANT SYSADM TO ROLE ADMINROLE;

RACF exit authorization – new CRITERIA keyword:
    PERMIT DSNADM SUBSYS.SYSADM ID(ADMINA)
       WHEN(CRITERIA(SQLROLE(ADMINROLE)))
```

Trusted context - Local

- Trusted context can be local or remote
- Local trusted context is based upon
  - System Authid
    - User ID associated with the connection
  - JOBNAME
    - Job or started task name associated with the connection

```
Example: Assign a role DBAROLE to any job named ADMINJOB that
connects using auth ID SALLY

CREATE ROLE DBAROLE;

CREATE TRUSTED CONTEXT DBACONTEXT
   BASED UPON CONNECTION USING SYSTEM AUTHID SALLY
   ATTRIBUTES JOBNAME(‘ADMINJOB’)
   DEFAULT ROLE DBAROLE
   ENABLE;
```

Trusted Context - Remote

- Remote trusted context is based upon
  - System Authid
    - User ID associated with the connection
  - ADDRESS
    - Client’s IP address, domain name or SERVAUTH security zone name of the connection
  - ENCRYPTION
    - Connection encryption level (NONE | LOW | HIGH)

```
Example: Assign a role TELLER to a connection established from
IP address 9.10.10.120 and the auth ID SRVRID01.

CREATE ROLE TELLER;

CREATE TRUSTED CONTEXT TELLERCONTEXT
   BASED UPON CONNECTION USING SYSTEM AUTHID SRVRID01
   ATTRIBUTES ADDRESS(‘9.10.10.120’)
   DEFAULT ROLE TELLER
   ENABLE;
```

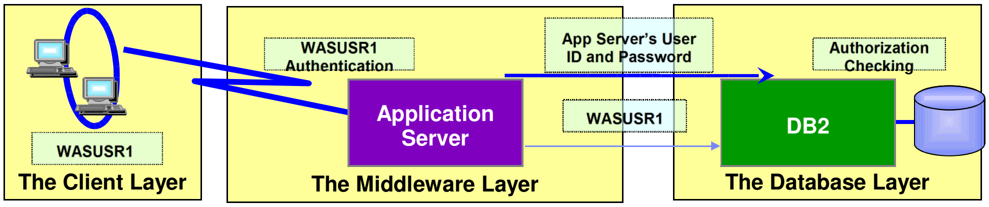
Trusted Context Auth ID Switching

- Allows trusted connection to be used by different users
- Optional authentication requirement
- Specific ROLE and RACF Security Label can be assigned to the user

```
Example: Assign a role TELLER to a connection established from
IP address 9.10.10.120 and the auth ID SRVRID01. Allow MARY and
JOHN to use the connection.

CREATE TRUSTED CONTEXT TELLERCONTEXT
   BASED UPON CONNECTION USING SYSTEM AUTHID SRVRID01
   ATTRIBUTES ADDRESS(‘9.10.10.120’)
   DEFAULT ROLE TELLER
   WITH USE FOR MARY, JOHN
   ENABLE;
```

Trusted Authentication with Identity Propogation



- The application server's user ID and password are used to establish the trusted connection.
- The user is switched in the trusted connection and client user ID is propagated to the server
- The client authorization ID's privileges are checked for database access

Trusted context: Secure DBA Activities

- Customers concerned about DBA access to sensitive data.
- An auditable DBA process can be done with trusted context and role:
  - Grant DBA privileges to a Role, AuditRole
  - When a DBA needs to perform a system change:
    - Create trusted context to assign AuditRole to a DBA auth ID
    - Enable trusted context to allow access to sensitive objects
    - DBA connects and performs activity against sensitive objects
    - Disable trusted context to protect sensitive objects
  - An auditor can review the audit trace

DB2 10: Row and Column Access Controls

- New data controls at the table level to protect against unplanned and dynamic SQL access
  - Can be defined with DB2 native authorization
- **Row Access control**
  - Establishes a row policy for the table to protect SQL access to individual rows
  - Defined as a row permission using SQL CREATE PERMISSION statement
- **Column Access control**
  - Establishes column policy for a table to mask column values in answer set
  - Defined as a column mask using SQL CREATE MASK statement

Audit

Auditing in DB2

- Who is privileged to access what data?
  - Most of the catalog tables describe the DB2 objects, such as tables, views, table spaces, packages, and plans
  - If using DB2 native authorization, several other tables (every table with the character string "AUTH" in its name) hold records of every granted privilege or authority.
- Who accessed what data?
  - You can find answers by using the audit trace, another important audit trail for DB2

DB2 Instrumentation Facility

- DB2 uses SMF and/of GTF and/or monitor program for trace data
- Trace types
  - Accounting
  - Audit
  - Monitor
  - Performance
  - Statistics
- Fully supported interface and extensive DB2 information
- Filtering capabilities that INCLUDE and EXCLUDE based on various keywords
  - Positioning and terminating wildcards can be used

Trace threads for all plans except plans that start with A, B and only where the user ID is USR1

```
-START TRACE (AUDIT) XPLAN(A*,B*) USERID(USR1)
```

Audit Trace Records

- Selective tracing with 11 classes of information
  - Access denials
  - Authorization changes
  - Changes to the structure of data (such as dropping a table)
  - Changes to data values (such as updating or inserting records)
  - Reading of data values (such as select)
  - Changes in authorization IDs
  - Utilities changes
  - Trusted context information
  - Audit Administrative Authorities

```
-START TRACE (AUDIT) CLASS (4,6) DEST (GTF) LOCATION (*)
```

DB2 10: Audit Policies

- Provide needed flexibility to audit any access to specific tables for specific programs during day
  - Does not require AUDIT clause to be specified using DDL
  - Generates records for all read and update access for statements with unique statement identifier
- Identify any unusual use of privileged authority
- Up to 8 audit policies can be specified to auto start or auto start as secure during DB2 start up
- Audit policy supports eight categories that maps to AUDIT classes.

```
INSERT INTO SYSIBM.SYSAUDITPOLICIES (AUDITPOLICYNAME,
OBJECTSCHEMA, OBJECTNAME, OBJECTTYPE, EXECUTE)
  VALUES ('TABADT1',‘EMPLOYEE','”PAY%’”','T','A');

-STA TRACE (AUDIT) DEST (GTF) AUDTPLCY(TABADT1);
```

Auditing with RACF exit authorization

- Failure SMF records written after entire list of profiles is exhausted
- SMF records have correlation information
- DB2 trace record IFCID 314
  - Traces all calls to the exit

DB2 Security provides

- Access control to DB2 objects using
  - DB2 Security
  - Access Control Authorization exit security
- Trusted context for better manageability and user accountability
- Row and column access control to safeguard data
- Enhanced auditing capability

References

- Security Functions of IBM DB2 10 for z/OS (SG24-7959-00)
  - **http://www.redbooks.ibm.com**
- DB2 10 for z/OS Technical Overview (SG24-7892-00)
  - **http://www.redbooks.ibm.com**
- DB2 10 for z/OS Managing Security (SC19-3496-01)
  - **http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.seca/src/seca/db2z_seca.htm**
- DB2 10 for z/OS Administration Guide (SC19-2968-02)
  - **http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.admin/src/admin/db2z_admin.htm**
- DB2 10 for z/OS RACF Access Control Module Guide (SC19-2982-02)
  - **http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.racf/src/racf/db2z_racf.htm**
- DB2 9 for z/OS: Configuring SSL for Secure Client-Server communications - Red paper
  - **http://www.redbooks.ibm.com/abstracts/redp4630.html?Open**
- DB2 10 for z/OS: Configuring SSL for Secure Client-Server communications - Red paper
  - http://www.redbooks.ibm.com/redpieces/abstracts/redp4799.html?Open
- DB2 for z/OS Information Center
  - **http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp**