Db2 12 for z/OS Security Updates

New England Db2 User's Group

June 24, 2021

Gayathiri (Gaya) Chandran, IBM

Agenda

Db2 12 for z/OS Security updates

Remote Security

Multi-factor Authentication

Encryption

Data at rest Encryption

Data in memory Encryption using Built-in Functions

Audit

Tamper-proof Audit Policies

Remote Security

What is Multi-Factor Authentication?

SOMETHING THAT YOU KNOW

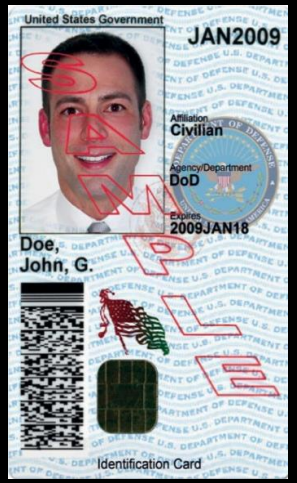
-Usernames and passwords

SOMETHING THAT YOU HAVE

- ID badge
- One-time passwords

SOMETHING THAT YOU ARE

- Biometrics



IBM Multi-factor Authentication (MFA) for z/OS

Extends RACF to authenticate users with multiple factors

Supports a wide range of authentication systems including

– IBM MFA with RSA SecurID

– IBM MFA with RADIUS

– IBM TouchToken

– And more …

MFA credentials are one-time use only

IBM Multi-Factor Authentication and Db2 for z/OS

Multi-factor Authentication (MFA) mostly transparent to Db2

MFA credentials sent as passphrase from remote clients

One-time use MFA credential prevents MFA from working properly with connections that replays password

- Sysplex workload balancing (WLB) enabled clients
- Clients that require multiple connections to perform a task

Db2 Sysplex Group Authentication for MFA

Support MFA credential replay by IBM Data Server Driver for Java or non-Java configured for Sysplex workload balancing or seamless fail over in a Db2 for z/OS data sharing system

- Requires subsystem parameter, AUTHEXIT_CACHEREFRESH set to ALL
- Requires ICSF to be active

Db2 non-Sysplex WLB clients for MFA

Support MFA for non-Sysplex WLB clients such as tools that require multiple connections to perform the set of tasks that make up an operation

- Requires subsystem parameter, AUTHEXIT_CACHEREFRESH set to ALL
- Requires ICSF to be active
- Requires a non-zero value for new subsystem parameter, MFA_AUTHCACHE_UNUSED_TIME
  - Specifies duration is seconds that MFA security credentials from a distributed client can remain unused in the Db2 global authentication cache
    - Acceptable values: 0, 120 - 7200
  - Data sharing Group scope
  - Requires Install SYSADM or SECADM authority for online update

Db2 for z/OS and MFA support

On successful authentication of the user, RACF returns an indication to Db2 if MFA was used

- Db2 caches a hash value of authentication token that contains multiple factors including connection environment information

Cached Authentication token purged:

- RACF commands, ALTUSER REVOKE, DELUSER performed against the connection user ID or user ID revoked due to excessive bad passwords
- Sysplex Group Authentication: MFA-based security credentials not periodically re-authenticated at least once every 2 hours with any member of the group by the client driver from the same IP address
- Non-Sysplex WLB clients: The same MFA-based security credentials not reused by the same client IP address within the time specified in the subsystem parameter, MFA_AUTHCACHE_UNUSED_TIME

Db2 for z/OS MFA Support Summary

Sysplex Group Authentication

- IBM Data Server Driver for Java and non-Java configured for Sysplex workload balancing or seamless failover
- Supports MFA and RACF passticket authentication
- Cached entry is valid for 2 hours without reuse
- APARs PI94236, PH21433

Support for non-Sysplex WLB clients

- Supports all DRDA that are non-Sysplex WLB enabled clients and REST API connections
- Supports data sharing and non-data sharing
- Cached entry valid time without reuse is controlled by subsystem parameter, MFA_AUTHCACHE_UNUSED_TIME
- APARs PH21341, PH37509

Remote Security Enhancements

APAR PH08188

- Enables DDF for SSL only connections
- PORT and SECPORT can be defined with identical values
- Location aliases subset can be defined with the same values for PORT and SECPORT

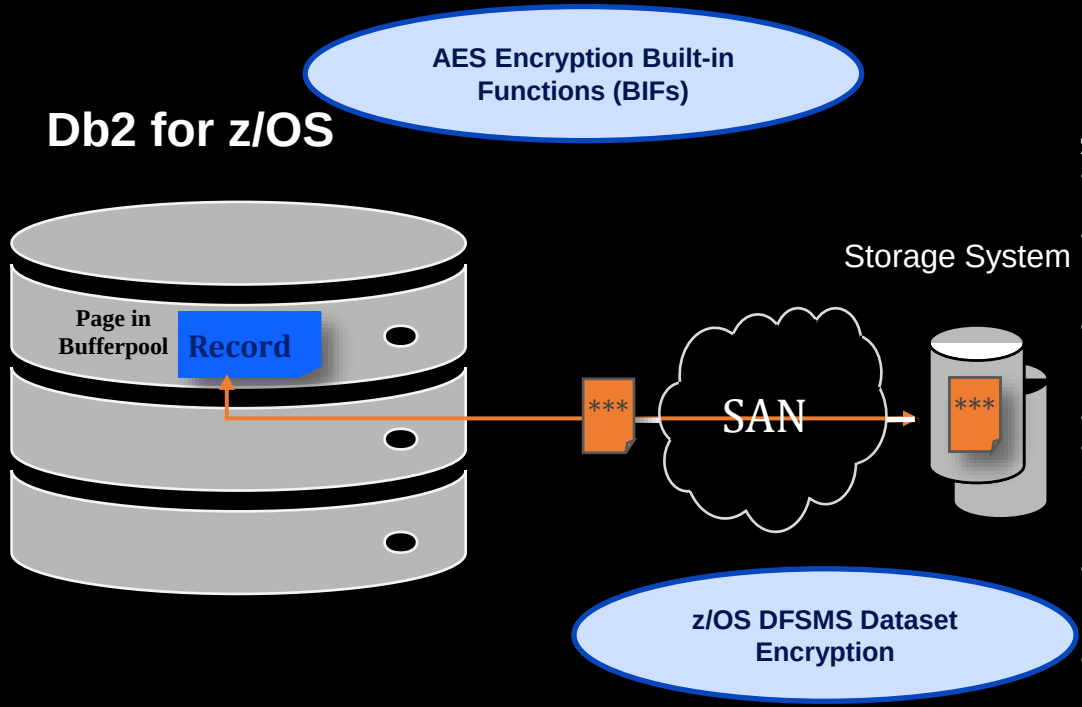
APAR PH16111

- IFCID 365 location statistics (DSNDQLST) entry, QLST
  - New fields are added to include number of connections using various authentication mechanisms, connection types, etc..
    - QLSTNREST QLSTNTRS QLSTNTLS QLSTNAES QLSTNMFA QLSTNCCA …

Encryption

Data Encryption Functions Overview

AES BIF Encryption:
- Data encrypted by ENCRYPT_DATAKEY BIF
- Data remains encrypted in the buffer pool and indexes
- Data decrypted by DECRYPT_DATAKEY_datatype BIF
- Secure application to view the data
- Supported in Db2 V12 FL505



z/OS Dataset Encryption
- Application transparent & enabled by policy
- Host encryption via CPACF as data written-to or read-from disk.
- Supports sequential and extended format VSAM data sets
- AES 256 bit key (XTS mode)
- Encrypt Db2:
  - ✓ Active & Archive log data sets
  - ✓ Catalog, directory & user table spaces
  - Supported in Db2 V11 and V12. V12 FL502 adds DBA controls.
  - Data in the buffer pool is not encrypted

DFSMS Policy-Based Dataset Encryption

Data sets are defined as encrypted by specifying a key label during the creation of a new data set

- SAF data set profile
- JCL, Dynamic Allocation, TSO Allocate, IDCAMS DEFINE
- SMS DATACLAS

During data set open, DFSMS:

- Checks the user access to the key label
- Specifies the key label to ICSF to retrieve the secure / protected key from the CKDS

ICSF fetches the secure key, uses the adapter to unwrap the key and CPACF to rewrap the key as protected key

- ICSF stores the protected key in ICSF cache

Steps to enable data set encryption

1

ICSF Admin

Generates an encryption key and key label. Stores it in the ICSF CKDS

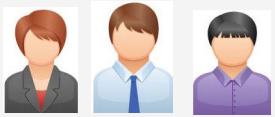
2

Security Admin

Sets up RACF for use of key label

- Allows secure key to be used as a protected key
- Permits **Db2 address space ID** use of the key label

3



Security / Database / Storage Admin

Associate the key label with the desired data set

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

(V12 only) In Db2, set key label for system objects and user objects

– OR –

In DFSMS, assign to data class

4

Database Admin

Migrate to encrypted data

Online REORG

Less disruptive

Encrypting Db2 System Objects (Logs, Catalog / Directory objects)

Options for key label specification

Security / Database System Admin / Storage Admin

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

In Db2, set key label using system parameter

– OR –

In DFSMS, assign to data class

Security Admin can set a key label in the RACF data set profile DFP segment using the new DATAKEY keyword

Application Database Admin can set a key label using ENCRYPTION_KEYLABEL system parameter (V12R1M502)

- Requires -SET SYSPARM command to take effect
- Data sharing Group scope
- Requires Installation SYSADM or SECADM authority to set the zparm
- Db2 DBM1 and MSTR address space IDs requires access to the key label

Storage Admin can set a key label using IDCAMS DEFINE for Active logs

Storage Admin can set a key label in the DFSMS data class

Encrypting Db2 System objects

Active logs

– Encrypt new active logs

- Define active log data set as encrypted and issue the SET LOG command NEWLOG option to add the newly defined active log data set to the active log inventory without stopping Db2

– Encrypt all active logs

- Stop Db2. Copy the contents of the active log data set to an encrypted data set. Restart Db2.

Archive logs

– New archive logs automatically encrypted based on the key label setting

Catalog and directory table spaces

– Execute REORG TABLESPACE utility to encrypt table spaces and index spaces in DSNDB06 and DSNDB01

– Encrypt DSNDB01.SYSUTILIX – Execute RECOVER utility followed by REBUILD INDEX ALL

Encrypting User Objects

Security / Database / Storage Admin

Options for key label specification

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

In Db2, set key label using SQL interfaces

– OR –

In DFSMS, assign to data class

Security Admin can set a key label in the RACF data set profile DFP segment using the new DATAKEY keyword

Application Database Admin can set a key label using SQL interfaces (APPLCOMPAT V12R1M502)

- CREATE TABLE
- ALTER TABLE
- CREATE STOGROUP
- ALTER STOGROUP

Storage Admin can set a key label in the DFSMS data class

Encrypting User Objects using Db2 controls at V12R1M502

SQL options for key label specification:

SQL CREATE / ALTER STOGROUP – New KEY LABEL option

- Adds a key label at the storage group level to encrypt all the table spaces using the storage group

SQL CREATE / ALTER TABLE – New KEY LABEL option

- Adds a key label at the table level to encrypt all the table spaces associated with the table
- Includes explicitly or implicitly created base table space, auxiliary table spaces, XML table spaces, index spaces
- Supported only for tables that reside in a universal table space or a partitioned table space

Encrypting User Objects

Execute the REORG utility to encrypt existing table spaces / index spaces

Additional utilities to convert to encrypted data sets:

- LOAD REPLACE, REBUILD INDEX, RECOVER from image copies
- REUSE option should not be specified
- PART or DSNUM option can be specified to encrypt / decrypt at the partition level

New table spaces or partitions defined are encrypted using the key label based on the hierarchy

Database Admin

Online REORG

Audit Readiness

Display encryption key label using DFSMS interfaces such as IDCAMS LISTCAT, SMF records

REPORT TABLESPACESET utility

- Displays key label associated for each catalog, directory and user table spaces using the new SHOWKEYLABEL option (V12R1M502)

DISPLAY LOG command

- Displays current key label information for current active log data sets (V12R1M502)

DISPLAY ARCHIVE command

- Displays current key label information for archive log data sets that are in use (V12R1M502)

Db2-supplied stored procedure, ADMIN_DS_LIST

- Displays data set encryption status and key label (APAR PH12920)

Utilities Consideration

All online utilities support table spaces and indexes whose underlying VSAM data sets are encrypted

Utility Input / Output data sets

– The options to define a key label for Utility data sets
  - RACF data set profile
  - JCL DSKEYLBL
  - SMS data class
– Authorization ID of the job requires access to the key label for any encrypted input or output data sets

Stand alone utilities

– Authorization ID of the job requires access to the key label for any encrypted data sets

Db2 for z/OS Encryption AES Built-in Functions

Encrypt data using ENCRYPT_DATAKEY function

Decrypt data using DECRYPT_DATAKEY_datatype function

When ENCRYPT_DATAKEY or DECRYPT_DATAKEY_datatype function is specified, Db2:

– Checks the user access to the key label

– Calls ICSF Symmetric Key Encipher callable service to encrypt the data

– Calls ICSF Symmetric Key Decipher callable service to decrypt the data

ICSF uses protected key for encryption and decryption

AES Encryption Built-in Functions

V12R1M505

ENCRYPTION: **ENCRYPT_DATAKEY** function

- Converts a block of plain text to a block of cipher text
- Uses the algorithm and the key label provided

```
>>--ENCRYPT_DATAKEY(data-string,key-label,algorithm)--><
```

Key Label

- A 64-byte label of the key in the ICSF CKDS that is used for the encryption / decryption
- Authorization is checked against the primary authorization ID of the process using the SAF CSFKEYS class

ENCRYPT_DATAKEY Built-in Function - Algorithms

Algorithms identify how the encrypted value can be searched

AES256D - Deterministic but no order

– 256-bit AES CBC mode encryption algorithm

– Uses a fixed Initialization Vector (IV)

– Equality operations can be performed on the cipher text

AES256R – Random

– 256-bit AES CBC mode encryption algorithm

– Uses a random Initialization Vector (IV)

– No range or equality operations can be performed on the cipher text

ENCRYPT_DATAKEY Built-in Function

Data type of the result is determined by the first argument as shown in the following table	
Data type of the first argument	**Data type of the result**
BIGINT, INT, DECIMAL, CHAR, VARCHAR, GRAPHIC, VARGRAPHIC	**VARBINARY**
CLOB, DBCLOB	**BLOB**

- A varying length header of a maximum length of 79 bytes included with the resulting encrypted value.

DECRYPT_DATAKEY Built-in Functions

V12R1M505

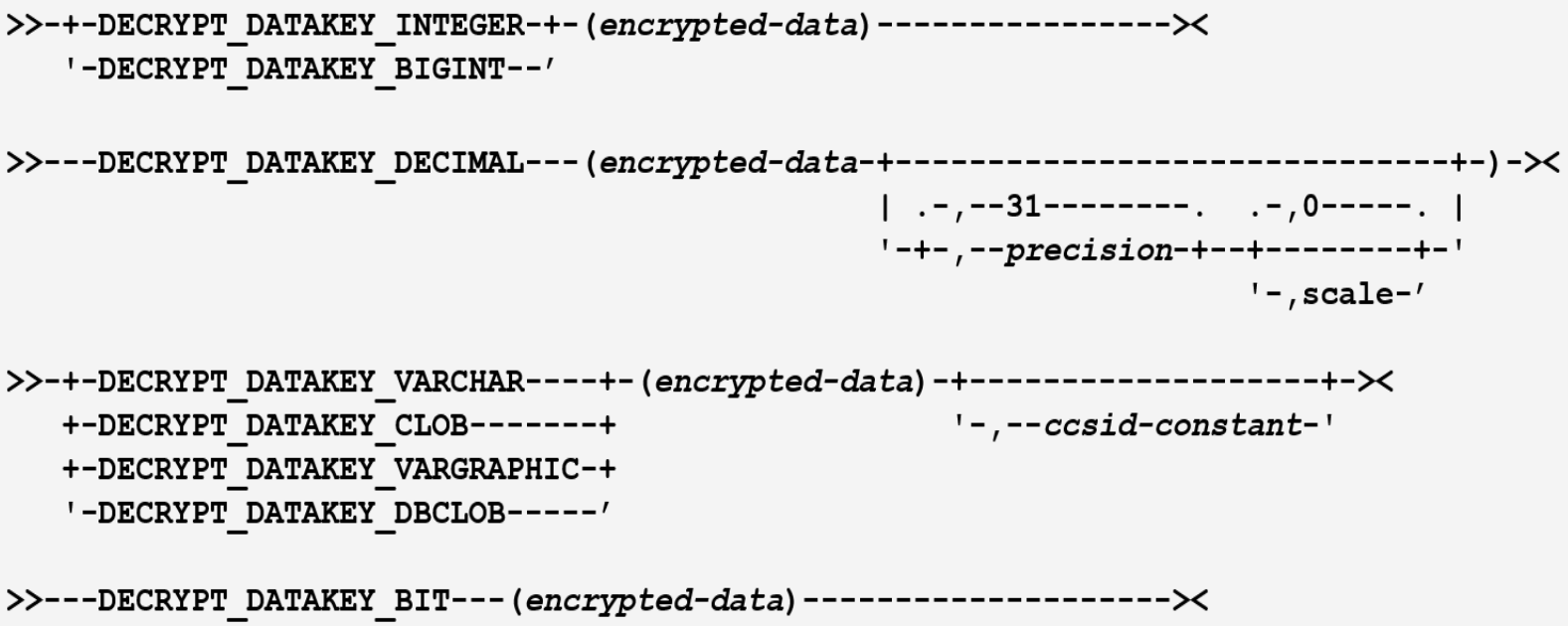
Returns a decrypted value using the algorithm and key label specified during encryption

Name of the function indicates the desired result data type

Eight DECRYPT_DATAKEY functions

- DECRYPT_DATAKEY_BIT
- DECRYPT_DATAKEY_CLOB
- DECRYPT_DATAKEY_DBCLOB
- DECRYPT_DATAKEY_DECIMAL
- DECRYPT_DATAKEY_BIGINT
- DECRYPT_DATAKEY_INTEGER
- DECRYPT_DATAKEY_VARCHAR
- DECRYPT_DATAKEY_VARGRAPHIC

DECRYPT_DATAKEY Built-in Functions



DECRYPT_DATAKEY Built-in Functions

Result of the decryption function			
Function	**Type of first argument**	**Actual type of encrypted data**	**Result**
DECRYPT_DATAKEY_BIGNT	BINARY, VARBINARY	BIGINT	BIGINT
DECRYPT_DATAKEY_BIT	BINARY, VARBINARY	CHAR, VARCHAR	VARCHAR FOR BIT DATA
DECRYPT_DATAKEY_CLOB	BLOB	CLOB	CLOB
DECRYPT_DATAKEY_DB CLOB	BLOB	DBCLOB	DBCLOB
DECRYPT_DATAKEY_DECIMAL	BINARY, VARBINARY	DECIMAL	DECIMAL
DECRYPT_DATAKEY_INTEGER	BINARY, VARBINARY	INT	INT
DECRYPT_DATAKEY_VARCHAR	BINARY, VARBINARY	CHAR, VARCHAR	VARCHAR
DECRYPT_DATAKEY_VARGRAPHIC	BINARY, VARBINARY	GRAPHIC,VARGRAPHIC	VARGRAPHIC

Steps to Enable Encryption using Built-in Functions

1

ICSF Admin

Generates an AES 256-bit encryption key and key label

2

Security Admin

Sets up RACF for use of key label

- Allows secure key to be used as a protected key
- Permits use of the key label to the application functional ID or group

3

Database Admin

Provides recommendation on the key label and the columns to be protected using the Built-in function

Performs necessary schema changes to store the results of the encrypt function

Analyzes the expected storage expansion, compression impact and any performance penalties

4

Application Programmer

Utilizes the new encryption built-in functions to protect columns with sensitive data

Data Encryption Summary

Data set Encryption provides various options to define a key label and migrate Db2 data sets with no application outages

New Built-in Encryption and Decryption functions make key management transparent for protecting data in memory

Make sure all disaster recover user ids and sites have access to any key labels used to protect Db2 encrypted objects and the key management system is fully deployed across the enterprise

Audit
Tamper-proof Audit Policies

Tamper-proof Audit Policies

Audit policies provide capability to define policies to audit various activities, such as privileged user access, access to tables, etc..

Tamper-proof audit policies prevents Db2 privileged users from modifying or stopping the audit policy

- Requires special task in a security product external to Db2, such as RACF

Tamper-proof audit policies

Audit policies is defined by inserting a record specifying the categories to be audited in SYSIBM.SYSAUDITPOLICIES table

SYSIBM.SYSAUDITPOLICIES column DB2START has a new value ‘T’ to identify tamper-proof audit policies

– Started automatically during Db2 start up

User needs additional privilege in RACF to modify or stop the audit policy

– Profile DSNAUDIT.*audit-policy-name* in DSNR class controls the tamper-proof audit policy access

Function Level, V12R1M509 must be activated to use the tamper-proof audit policies

For more information …

IBM Z Multi-Factor Authentication
- https://www.ibm.com/support/knowledgecenter/SSNR6Z
- https://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/seca/src/tpc/db2z_enablesysplexgroupauth.html
- https://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/seca/src/tpc/db2z_enablecachemfa.html

Data set Encryption:
- https://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/seca/src/tpc/db2z_dfsmsencryptionsupport.html
- https://www.redbooks.ibm.com/redpieces/abstracts/sg248410.html

AES encryption Built-in function
- https://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/seca/src/tpc/db2z_definecol4encryptdatakey.html

Blogs on Db2 encryption support:
- https://www.idug.org/browse/blogs/blogviewer?blogkey=c0410888-6589-4bf4-b7f5-589108ac680b
- https://community.ibm.com/community/user/hybriddatamanagement/blogs/paul-mcwilliams1/2019/12/09/encrypting-a-column-in-db2-without-modifying-the-a

Thank you

Gayathiri Chandran
gchandran@us.ibm.com