Best practices: Step by step instructions to configure a secure database system

Greg Stager

IBM Canada

Tridex
A DB2 USER GROUP

Agenda

Tridex
A DB2 USER GROUP

- Learn the best practices:
  - Required for OS configuration to ensure a secure Db2 setup
  - Authenticating users to Db2
  - Controlling what data users have access to through various authorization features of Db2
  - Encrypting both data at rest and data in motion
  - Tracking database activity using audit

Our Goals

Tridex
A DB2 USER GROUP

- Practical advice for a novice or intermediate database administrator to setup an initially secure Db2 server
- Focus on the “biggest bang for the buck” items
- Best practices listed are intended as a starting point
  - Advanced setups my differ from those listed here

System setup details

Tridex
A DB2 USER GROUP

- To limit our talk to 1 hour, we will make some simplifying assumptions
- Single server version of Db2
  - not DPF or pureScale
- RHEL 7

Tridex
A DB2 USER GROUP

Learn the best practices required for OS configuration to ensure a secure Db2 setup.

The server for Db2

Tridex
A DB2 USER GROUP

- Use a dedicated server for Db2
  - Files like db2diag.log are world readable
  - Want strong control over who can log into the server, only administrators
  - Historically we've had more security vulnerabilities that can only be exploited by users who can log into the OS than those that can be exploited remotely
- Don't change Db2 installed file permissions
  - This can often lead to unexpected behaviour
- Only instance owner needs access to database path and transaction logs

Configure LDAP authentication

Tridex
A DB2 USER GROUP

- Make sure necessary packages are installed

```
yum install -y  openldap-clients sssd authconfig sssd-client
```

- Enable SSSD and LDAP

```
authconfig --enableshadow --passalgo=sha512 --enablesssd --enablesssdauth --
enableldap --enableldapauth --enableldaptls --ldapserver=”<hostname>" --
ldapbasedn=”<basedn, o=…>" --update
```

- Download your Root CA cert for the LDAP server
  - Add to ‘/etc/openldap/cacerts/’
- Restart SSSD

```
systemctl restart sssd.service
```

Control Who Can Login to the server

Tridex
A DB2 USER GROUP

- Make a copy of the pam file for later use (before it gets changed)

```
cp /etc/pam.d/system-auth /etc/pam.d/db2
```

- Modify /etc/security/access.conf

```
+ : root wheel : ALL
+ : db2inst1 : ALL
- : ALL : ALL
```

- Modify /etc/sysconfig/authconfig
  - Change the following line to yes

```
USEPAMACCESS=yes
```

- Run `` `authconfig --updateall` ``

Create instance accounts

Tridex
A DB2 USER GROUP

- Have root create instance owner and fenced mode user

```
groupadd  db2iadm1
groupadd  db2fsdm1
useradd -g db2iadm1 db2inst1
useradd -g db2fsdm1 db2fenc1
passwd db2inst1
passwd db2fenc1
```

- These names are common, but not required
  - Limited to 8 characters, not many special characters supported

Install Db2

- Perform a typical/default install of Db2 as root

```
./db2_install -p SERVER
```

- Create instance

```
./db2icrt -s ese -u db2fenc1 db2inst1
```

Add database users to OS if not using LDAP

Tridex
A DB2 USER GROUP

- useradd <username>
- passwd <username>

- Do not add these users to access.conf, you don’t want them logging into the OS

Where to go next with OS setup

Tridex
A DB2 USER GROUP

- Firewalls
  - Typically only need the port used for TLS (SSL_SVCENAME) open, plus SSH
- Other typical server hardening

Tridex
A DB2 USER GROUP

Learn the best practices for securely authenticating users to Db2

Authentication Types

- Authentication is the act of checking your proof of identity
  - Abbreviation AUTHN
- The AUTHENTICATION parameter in the Database Manager Configuration determines which security mechanism Db2 uses for authentication
- Users are always defined externally to Db2

Example authentication types:

- SERVER_ENCRYPT
- SERVER
- DATA_ENCRYPT
- KERBEROS
- GSSPLUGIN
- TOKEN_SERVER_ENCRYPT

Example user definition locations

- Operating System
- LDAP
- Kerberos
- Plugin
- Identity Provider Token

Best Practice - Use SERVER_ENCRYPT

Tridex
A DB2 USER GROUP

- Use SERVER_ENCRYPT
  - Encrypts usernames and passwords sent during connect
  - Supports local users or LDAP
- Do NOT use CLIENT
  - Anyone with a network connection can impersonate any other user
- Do NOT use DATA_ENCRYPT
  - Deprecated, only supports DES encryption
  - Use TLS instead

Using SERVER_ENCRYPT securely

Tridex
A DB2 USER GROUP

- Default is DES encryption, but we really want AES
  - set ALTERNATE_AUTH_ENC to AES_ONLY in DBM CFG

- For historical compatibility reasons, SERVER_ENCRYPT does not enforce encryption for JDBC connections
  - You can force its use
    - db2set DB2AUTH=JCC_ENFORCE_SECMEC
  - Or log its abuse
    - db2set DB2AUTH=JCC_NOENFORCE_SECMEC_MSG

Transparent LDAP

- The most popular authentication method used is SERVER_ENCRYPT with Transparent LDAP enabled
  - db2set DB2AUTH=OSAUTHDB
  - Db2 makes OS calls via PAM APIs (Linux), OS in turn looks locally or in LDAP
  - LDAP usage is transparent to Db2 as it is handled by OS
  - Even if not using transparent LDAP right now, you can still ‘future-proof’ your instance by changing this now

- LDAP plugins require ALL users to be defined at LDAP server
  - Many customers want instance owner and FMP user defined locally

Transparent LDAP Configuration (1/2)

- Db2 requires its own PAM configuration file in /etc/pam.d/db2
- In a previous step we made a copy of the system PAM configuration
  - We do not want the rule that limits access in the Db2 config, otherwise users will not be able to login unless they can log into the OS
  - Make sure this line is not present:
    - account required pam_access.so

Authentication Cache

Tridex
A DB2 USER GROUP

- New feature in Db2 11.5.3.0
- Improve performance for slow authentication and group lookup
- Db2 maintains cache of
  - Hashed password
  - Group membership
- Configured at database level
  - AUTHN_CACHE_USERS
    - How many users are in the cache (controls how much memory is used)
  - AUTHN_CACHE_DURATION
    - How long a cached entry is valid for
    - Expired entries will force authentication next time the user connects

Where to go next with authentication

Tridex
A DB2 USER GROUP

- Advanced Authentication Types
  - Kerberos
    - Single Sign-on support
  - JWT
    - Token authentication when integrating with an identify provider and web application
  - LDAP Plugins
    - Db2 makes LDAP API calls directly to an LDAP server
  - Custom GSSAPI plugins
    - You can create your own authentication code (in-depth C programming required)
- SRVCON_AUTH
  - Separate local authorization from incoming connect authentication

Tridex
A DB2 USER GROUP

Learn the best practices for controlling what data users have access to through various authorization features of Db2

Authorization - terminology

- Authorization is the go/no go decision of whether an action can take place
  - Abbreviation AUTHZ
- Authorities are collections of permissions centered around a related topic
  - Ex. SYSADM, DBADM, SECADM
- Privileges are individual permissions on specific objects
  - SELECT on a TABLE T1

Summary of authorities

Tridex
A DB2 USER GROUP

- Authorities are hierarchical, allowing delegation of common tasks

- SYSADM → SYSCTRL → SYSMAINT → SYSMON
- SECADM → ACCESSCTRL
- DATAACCESS
- DBADM → SQLADM → EXPLAIN
  → WLMADM

Which authorities to use

Tridex
A DB2 USER GROUP

- There are a lot of authorities - should you use them all?
  - The more they are held by distinct users the better, but given our goal of a practical setup for smaller cases, then no, don't try to use them all
- Database creator gets SECADM, ACCESSCTRL, DATAACCESS, DBADM
- First separation is to focus on is having SECADM and DBADM held by different users
  - Numerous advanced security functions are only usable by SECADM
- Secondly, eliminate the use of DATAACCESS
  - Watch out, DATAACCESS is granted by default with DBADM
  - Be explicit in all the permissions you grant

Authority Configuration

Tridex
A DB2 USER GROUP

- These recommendations leave us with two distinct administrators
  - Database Administrator
    - SYSADM
    - DBADM
    - ACCESSCTRL (to handle day to day grants/revokes)
  - Security Administrator
    - SECADM - advanced security functionality and auditing of Db admins

- The database creator has had DATAACCESS revoked by SECADM

Numerous privileges are granted to PUBLIC during create database

- There’s a few that should be removed
  - CONNECT
    - Normally Db2 can authenticate a wider range of users than should be connecting to your database - for example everyone in your LDAP server
  - IMPLICIT SCHEMA
    - Implicit schemas (no CREATE SCHEMA statement) are owned by the system and PUBLIC can create objects in it
    - Users should be explicitly creating schemas (DBADM has implicit IMPLICIT_SCHEMA)
  - CREATETAB
    - Most users have no need to create tables, this should be a controled activity
  - BINDADD
    - Most users have no need to create packages

Several privileges to make sure you do not grant to PUBLIC

- A few privileges may not be so obvious to strongly restricted:
- CREATE_EXTERNAL_ROUTINE
  - The ability to create C and Java routines that are run at the server
  - Given the broad capabilities of C/Java code, these must be restricted
- CREATE_NOT_FENCED_ROUTINE
  - Not fenced (aka trusted) routines run outside the Fenced Mode Process (FMP) sandbox and instead directly inside the Db2 server
  - It’s very easy for these routines to crash or corrupt the Db2 server when written in C or Java

Views for authorization delegation

Tridex
A DB2 USER GROUP

- Views can be used to control what data users see
- A user can be granted SELECT on the view without having access to the underlying base table(s) and other objects
  - View definer needs the access, but not the view user
  - SECADM and ACCESSCTRL can grant the SELECT
  - View DEFINER is given CONTROL, which includes the ability to GRANT on the view, if they had CONTROL on the base table(s) or DBADM/DATAACCESS
- Users with DATAACCESS can always select directly from the base table, so a view will not protect against these users
  - Otherwise views present a useful security mechanism

Routines for authorization delegation

Tridex
A DB2 USER GROUP

- Routines can modify and return data to the callers
- A user can be granted EXECUTE on a routine without having access to the underlying tables and other objects
  - Routine definer needs the access, but not the routine caller
    - For SQL routines. Dynamic and non-SQL routines are more complex
  - Routine definer given EXECUTE WITH GRANT
  - SECADM and ACCESSCTRL can also grant EXECUTE
- Allows you to encapsulate business logic into the routine and control access at the routine level

Using roles and groups to ease authorization maintenance

Tridex
A DB2 USER GROUP

- An application user may require access to dozens or more objects
- If you can define your users according to their job, you can grant privileges and authorities to roles or groups representing those jobs
- Grant the user membership in the appropriate role or group
- If a user changes jobs, it’s simple to remove them from the role or group instead of individual objects

Groups vs Roles (1/3)

Tridex
A DB2 USER GROUP

- For certain objects, Db2 records a dependency of the owner's privilege to access dependent objects
  - Ex. a user having SELECT on a TABLE to create a VIEW
- The user must maintain those privileges
- If the user loses those privileges, the objects will be marked as invalid or inoperative
  - Losing SELECT on the TABLE will make the VIEW inoperative
- Special case for users who hold DATAACCESS when the object is created
  - Dependency is not recorded

Affected Objects:

- Views
- Materialized Query Tables (MQTs)
- SQL routines
- Triggers
- Packages containing static SQL

Groups vs Roles (2/3)

Tridex
A DB2 USER GROUP

- For creation of listed objects, Db2 does not consider privileges obtained through groups
  - Db2 is not immediately aware of group changes in order to invalidate objects

- Roles alleviate this problem, Db2 **will** consider privileges from roles for these cases
  - Except for roles obtained through groups

Groups vs Roles (3/3)

Tridex
A DB2 USER GROUP

- Using LDAP for groups?
  - If possible, use roles for administrators who will be creating objects
    - Otherwise you must grant privileges to individual users for object creation
  - Use groups for application users

- No LDAP
  - Use roles for any in-database privileges
  - Still need groups for SYS* authorities at the instance level

Validate a user's authorities

- SYSPROC.AUTH_LIST_AUTHORITES_FOR_AUTHID table function
  - List the instance and database authorities held by a user
  - Shows if they are direct, through a group or role

```
SELECT *    FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('MYUSER', 'U') ) AS T

AUTHORITY                    D_USER D_GROUP D_PUBLIC ROLE_USER ROLE_GROUP ROLE_PUBLIC D_ROLE
---------------------------- ------ ------- -------- --------- ---------- ----------- ------
ACCESSCTRL                   N      N       N        N         N          N           *
CONNECT                      N      N       Y        N         N          N           *
DATAACCESS                   N      N       N        N         N          N           *
DBADM                        Y      Y       N        N         N          N           *
SECADM                       Y      N       N        N         N          N           *
SYSADM                       *      Y       *        *         *          *           *
…
```

Check what database authorities have been granted

- The SQL in the speaker notes will show database authorities that are held by:
  - Users, groups, roles, PUBLIC
  - Also via nested roles: ex. DBADM granted to role1 granted to use U3

AUTHORITY	GRANTEE	GRANTEETYPE	VIA	VIATYPE
DBADM	U1	U	-	-
DBADM	G1	G	-	-
DBADM	R1	R	-	-
DBADM	U3	U	R1	R

Check what privileges have been granted

- The view SYSIBMADM.PRIVILEGES is a summary of all the privileges stored in the catalog tables (it's a big UNION ALL statement)

```
SELECT AUTHID, PRIVILEGE, OBJECTNAME, OBJECTSCHEMA, OBJECTTYPE
   FROM SYSIBMADM.PRIVILEGES

AUTHID       PRIVILEGE   OBJECTNAME                 OBJECTSCHEMA     OBJECTTYPE
--------...- ---------- ------------------...- -----------...- ----------
...-
GSTAGER      CONTROL     EMPLOYEE                   GSTAGER          TABLE
GSTAGER      ALTER       EMPLOYEE                   GSTAGER          TABLE
GSTAGER      DELETE      EMPLOYEE                   GSTAGER          TABLE
GSTAGER      INSERT      EMPLOYEE                   GSTAGER          TABLE
GSTAGER      SELECT      EMPLOYEE                   GSTAGER          TABLE
GSTAGER      UPDATE      EMPLOYEE                   GSTAGER          TABLE
```

Look at individual catalog views for targeted details

Some common SYSCAT views

- DBAUTH
- TABAUTH
- SCHEMAAUTH
- ROUTINEAUTH
- <OBJ>AUTH

Where to go next with authorization

Tridex
A DB2 USER GROUP

- CREATE DATABASE RESTRICTIVE
  - Eliminates all grants to PUBLIC at create db time and in the future
  - Due to Db2's use of packages for SQL execution from CLP, CLI etc, a difficult task to initially get setup
- Separation of Duties
  - Ideal goal is the user who grants authorities is different than those who can access data
  - Different user for SECADM, ACCESSCTRL, DBADM etc.

Where to go next continued

- Row and Column Access Control
  - Column masking and row permissions
  - Custom SQL rules to control what users have access to
- Label Based Access Control (LBAC)
  - Complicated access control rules based on labels
  - RCAC is much more user friendly
- Trusted Context
  - Rules that define a trusted connection, which allows
    - Dynamic access to a role only within that connection
    - Ability to switch to other users (for middle tier applications to preserve end user identity)

Tridex
A DB2 USER GROUP

Learn the best practices for encrypting both data at rest and data in motion

TLS - Transport Layer Security (1/3)

Tridex
A DB2 USER GROUP

- TLS provides encryption of data in motion, over the network
  - Any references to SSL are synonyms for TLS
- Configure it in place of TCPIP for client-server communication
  - Don’t forget to use it for HADR as well!

TLS - Transport Layer Security (2/3)

- Unfortunately, the default version of TLS that Db2 uses is TLS 1.0 and 1.1, which are insecure with known vulnerabilities
- You must configure the use of TLS 1.2
- set SSL_VERSIONS to TLSV12 in dbm cfg

TLS - Transport Layer Security (3/3)

- Make sure keystore files are secure at client and server
  - Access to private key at server could allow someone to masquerade as server
  - Access to keystore file at client could allow accepting rogue CA signed certificates (fake servers)
- File should only be readable/writeable by:
  - Server - the instance owner
  - Client - the application
  - These are the defaults, don’t change them

Native Encryption (1/3)

- Db2 Native Encryption provides built in, application transparent encryption of data at rest (on disk) for:
  - Database container files
  - Transaction Logs
  - Backup files

- Protects against *offline* attacks against data
  - accessing the data outside of the database manager

Native Encryption (2/3)

- Enabling native encryption is very easy
  - ENCRYPT keyword on CREATE DATABASE or RESTORE
- Key management is difficult
  - Administrative challenge to manage keys
  - Keys must be backed up, maintained for long periods
  - You can end up with numerous keys
    - Each database should have its own key, they can also be rotated
- Enable native encryption if you can manage the administration
  - Failure can mean complete loss of data if un-backed up keys are lost

Native Encryption (3/3)

- Similar to TLS, you need to ensure keystores are protected
  - Only the instance owner should have read/write access
- Make sure your keystores are backed up!!!
  - Also make sure you know the password to open those keystores

Where to go next with native encryption

Tridex
A DB2 USER GROUP

- Switch from keystore files to centralized key managers
  - KMIP servers are most popular
    - Similar to LDAP server for keys
  - Hardware Security Modules provide ultimate in key security
    - Often more difficult to setup and use
  - Provide central management of keys, backup, access control etc.

- Investigate how keys are different for databases and backups
  - Default is the use the same key, but they can be different
  - Useful if you want to restore the database somewhere, but don’t want to expose the live key for the database (test system etc).
  - You can also encrypt one or the other (default is both)

Tridex
A DB2 USER GROUP

Learn the best practices for tracking database activity using audit

Audit

- Db2 has a built-in audit facility that allows you to track actions within the database
- Two levels of configuration are provided to fine tune what is audited

Categories
- AUDIT - use of audit facility
- CHECKING - authorization checks
- SECMAINT - grants/revokes
- OBJMAINT - object creation/deletion (some alters)
- CONTEXT - contextual information for other events
- EXECUTE - SQL statement execution
- SYSADMIN - SYS* actions
- VALIDATE - authentication checks

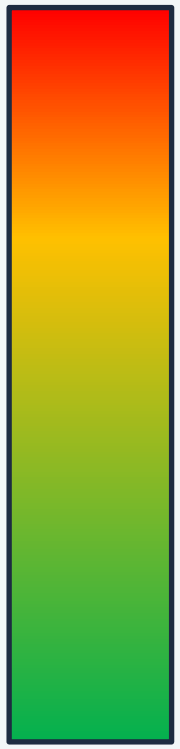
Objects to Audit
- Database
- Users/Groups/Roles
- Authorities
- Tables
- Trusted Contexts

Impact of Audit

Tridex
A DB2 USER GROUP

- Auditing *everything* can have a substantial impact to performance on a busy OLTP system
  - AUDIT_BUF_SZ in the dbm cfg can be configured to allow buffered writes of audit events drastically improving performance
- The amount of data generated by auditing *everything* can be overwhelming on a busy OLTP system
  - Gigabytes per minute
- Audit only what you need

Which categories to audit



- EXECUTE
- CONTEXT
- CHECKING
- VALIDATE
- SYSADMIN
- OBJMAINT
- SECMAINT
- AUDIT

- AUDIT, SECMAINT, OBJMAINT and SYSADMIN events occur infrequently enough they can generally always be audited
- CHECKING and VALIDATE have a medium impact, audit failures
- EXECUTE and CONTEXT have a high impact, need to be very targeted

Audit Recommendations

Tridex
A DB2 USER GROUP

- Audit everything at the instance level
- Audit the database for AUDIT, SECMAINT, OBJMAINT, SYSADMIN
- Audit failures at the database for CHECKING, VALIDATE
- Audit admins for everything

Analyzing Audit Output

Tridex
A DB2 USER GROUP

- Db2 writes to the active log file, which is then archived (copied)
- Only the instance owner should have read/write permissions on these files
  - Can contain SQL statements and input data, may be considered sensitive
- Output is in three formats:
  - text based report
  - CSV suitable for db2load
  - syslog
- Best practice is to analyze the output on a different system
  - Don’t want admins able to hide their tracks by modifying the audit log in tables

Where to go next with audit

- Previous recommendations audit as much as possible without a large system impact
  - You may have additional goals requiring more auditing
- CHANGE HISTORY EVENT MONITOR can provide additional insight into database activities
- There is no guidance on analyzing audit output
  - 3rd party products such as IBM Guardium Data Protection for Databases can provide advanced tooling and analysis

Stay up to date with fixes!

Tridex
A DB2 USER GROUP

- 60% of data breaches are the result of unpatched systems
- Any security related APAR is accompanied by a security bulletin
  - Describes at a very high level the impact of the vulnerability
  - Vague on purpose, keep details out of the hands of hackers
- See the list of published security bulletins
- Security fixes are published simultaneously across all supported releases as either fixpacks or special builds
- Subscribe to My Notifications to find out about new bulletins

Advanced Hardening Actions

Tridex
A DB2 USER GROUP

- Db2 STIG
  - Security Technical Implementation Guides by DoD
- Centre for Internet Security Benchmark for Db2
  - Guidelines for securing a Db2 Server
  - Updates coming soon for V11.5
  - Community driven effort, feel free to participate!
- Both are a little old, but still provide good practical advice

- Guardium Vulnerability Assessment
  - Detect vulnerabilities and misconfigurations in your database server
  - Encodes rules from STIG and CIS guides

Greg Stager
IBM Canada
gstager@ca.ibm.com