DB2 Connect Virtual Users Group

DB2 Connect and DB2 for z/OS Security

DB2 Connect

Hosted and moderated by:

- Frank Fillmore, The Fillmore Group

THE FILLMORE GROUP
Relational Database Solutions

DB2 Connect

Housekeeping

Today's session is scheduled to run for 90 minutes.

All attendees' microphones will be muted for the duration of the webcast.

Presentation slides will be posted to The Fillmore Group blog after the presentation (www.thefillmoregroup.com/blog)

We are attempting to record this session. If a replay is available we will notify today's session attendees.

If you or a colleague would like to receive emails with information on upcoming DB2 Connect Virtual Users Group sessions please email Kim May, kim.may@thefillmoregroup.com

Many thanks to the volunteers who contribute to the preparation and delivery of these sessions.

THE FILLMORE GROUP
Relational Database Solutions

Upcoming DB2 Connect Classes

- CF632 - **DB2 Connect 9 for DB2 for z/OS Problem Determination and Performance**

  Details: Cost = $2600, Duration = 4 days, 1 session scheduled to begin 11/14 in Towson, MD

- CL600 - **Implementing DB2 Connect and Client Connections to DB2 for z/OS**

  Details: Cost = $1300, Duration = 2 days, sessions scheduled to begin:

- 05 Dec 2011 Towson/Baltimore, MD
- 27 Feb 2012 Towson/Baltimore, MD
- 02 Apr 2012 Durham/Raleigh, NC

DB2 Connect Training

Presented by:

- Adam Koile
- Mary Ann Ritosa

Let's Get Started

DB2 Connect and DB2 for z/OS Security

IBM Software Group

DB2 Connect & DB2 z/OS Security

Worldwide Information Management

***By Adam Koile & Mary* Ann Ritosa**

ON DEMAND BUSINESS

Objectives & Requirements

Objective: We will discuss security concepts involved for both DB2 UDB v9.1, v9.5, v9.7 and DB2 z/OS

Outline:

1. What is security?
2. What configuration parameters are involved?
3. What are the possible configurations?
4. Special considerations
5. Windows Domain users
6. LDAP and Transparent LDAP
7. DB2 Z/OS
8. References

What is security?

- The topic of security covers both DB2's use of external security services.
- When attempting to access a database server, you must first identify yourself and provide proof of your identity. This is called "authentication". The proof of your identity is in the form of an authentication token, which in many cases will be a password.
- Once you have been successfully authenticated, the next step is to determine the user's authorities. "Authorization" is the process of verifying that a userid is allowed to perform a particular action, or access specific data.
- For a connection or attachment to succeed, both authentication and authorization must be successful.

What is security?

- DB2 relies completely upon external security facilities to accomplish user authentication. The security facility can be part of the operating system, or a separate product. This leverages new developments in security and reduces the administrative overhead of requiring users and groups to be defined to the database. DB2 relies on the configured security facility to provide both user and group membership information, and thereby subjects the DB2 users to the same restrictions imposed by the security facility. This allows DB2 to interoperate with native operating system security, taking advantage of the many solutions provided there, or to look to external third party security mechanisms such as LDAP with no administrative overhead within the database.
- Once a user has been successfully authenticated by the system, DB2 records the user's identity and other relevant security information, which is kept for the life of the session. Internally, the user is known to the system using an authorization identifier or "authid". This name can be the same as the user ID. As an example, on UNIX based systems, a DB2 *authid* is derived by transforming to uppercase letters a UNIX user ID that follows DB2 naming conventions. Within the DB2 system, the user will be referred to by this authid from this point onwards.

What is security?

- It is important to note that there are rules for the naming of all objects, including users. Some of these rules are specific to the platform you are working on. For example:
  - On UNIX platforms, user names can contain up to 8 bytes and must be in lower case.
  - On Windows platforms, user names can contain up to 30 characters, and can be in upper, lower, and mixed-case. Windows currently have a practical limit of 30 characters.
  - Group names can contain up to 8 bytes.
  - Names and IDs cannot be USERS, ADMINS, GUESTS, PUBLIC, LOCAL or any SQL reserved word or begin with IBM, SQL or SYS.
- Password rules
  - Linux, Uinx and AIX 5.3 (lower) 8 characters or less in v9.1
  - AIX 6.1 and higher 64 characters or less for MD5 or SHA1 compliant in v9.1
  - Windows 14 characters on v9.1
  - In v9.5 and v9.7 password length can be up to the maximum number of characters supported by your operating system.
- When authentication has successfully completed, DB2 will also have obtained a list of the groups to which the user belongs. Group membership can then be used to authorize the user.

What configuration parameters are involved?

- How you specify the authentication type depends on what protocol you are using, and what role you are playing in the environment, i.e. client, gateway or server.
- If you are acting as the server, the authentication type is always determined by the AUTHENTICATION parameter in the database manager configuration file associated with the instance.
- If you are acting as the client, the authentication type can be specified in the database catalog, but this setting may also be affected by the communication protocol used to connect to the server.
- If you are acting as the gateway, the authentication type is specified in a combination of both the aforementioned manners.

Example 1 – Setting up the server

- As an example, let's set up a client – server environment. On the server, log in as a system administrator, and check the authentication type in use:

❑ **db2 get dbm cfg |more**

- If it is not already set to SERVER, please do so now as follows:

❑ **db2 update dbm cfg using authentication SERVER**

❑ **db2stop**

❑ **db2start**

```
SYSADM group name                          (SYSADM_GROUP) = USR
SYSCTRL group name                        (SYSCTRL_GROUP) =
SYSMAINT group name                      (SYSMAINT_GROUP) =

Database manager authentication          (AUTHENTICATION) = SERVER
Cataloging allowed without authority     (CATALOG_NOAUTH) = NO
Trust all clients                        (TRUST_ALLCLNTS) = YES
Trusted client authentication            (TRUST_CLNTAUTH) = CLIENT
Bypass federated authentication              (FED_NOAUTH) = NO
```

Valid authentication types

- The following is a list of the currently supported authentication types:

Authentication	Required Credentials	Where authentication occurs
SERVER	Password (not required for local connections)	Server
SERVER_ENCRYPT	Encrypted password *	Server
SERVER_ENCRYPT_AES	Send both the user ID and password encrypted. Advanced Encryption Standard (AES)	Server
CLIENT	Password (optional)	Client (depending on the values of trust_allclnts and trust)clntauth parameters
DATA_ENCRYPT	All data being sent is encrypted	Server

* An unencrypted password is sufficient if the client has specified an authentication type of SERVER in its database catalog. This was allowed to support down-level clients that did not support encryption.

Example 2 – Setting up the client

- The same authentication types are supported on the client, however the method of setting them differs. On the client, it is optional to specify the authentication type, and the method of doing so is via the CATALOG DATABASE command.
- As an example, configure the following on your client:

❑ **db2 catalog tcpip node <nodename> remote <hostname> server <port number>**

❑ **db2 catalog database SAMPLE as <db alias> at node <nodename> authentication SERVER**

❑ **db2 terminate**

❑ **db2 list db directory** {To view the results.}

- You should now be able to connect successfully to the server from your client, specifying a userid and password that exist on the server, as follows:

❑ **db2 connect to <db alias> user <userid>**

- If the connection fails, ensure that the following has been done on the server, then retry:

➢ db2set DB2COMM=TCPIP

➢ The dbm cfg parameter SVCENAME has been set, and has a value that equates to the <port number> specified above.

➢ The instance has been started

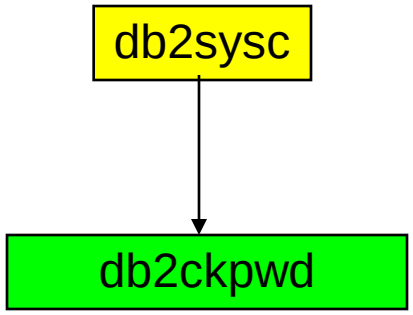
Why specify client authentication type?

- If the AUTHENTICATION type is optional on the client, why would you want to set it? Unless the database resides on a host, most customers do not indicate an expected authentication type, i.e., the client's authentication is not specified. The two main arguments against this are as follows:
  - Improved performance. If the client has the correct authentication type cataloged, the needed authentication tokens will be sent in the initial connection request.
  - There is no ambiguity for the user as to the type of authentication used, since it is clearly documented in the client's database catalog.

- If the client has not specified an authentication type, DB2 will attempt to use SERVER_ENCRYPT by default. If this type is not accepted by the server, the client will attempt to retry using an appropriate value returned from the server. If the server supports multiple authentication types, the client will not choose amongst them but will instead return an error. This is done to ensure that the correct authentication type is used. NOTE: Mainframes may return multiple authentication types since they can support them all simultaneously. It's possible that a host could return CLIENT, SERVER, SERVER_ENCRYPT, and Kerberos..

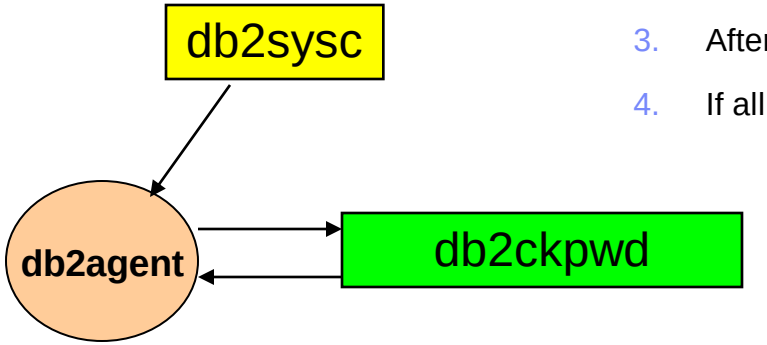
- If an authentication type is specified, then authentication can begin immediately provided that the value specified matches that of the server. If a mismatch is detected, then DB2 attempts to recover. Recovery may result in more flows to reconcile the difference or in an error if DB2 cannot recover. In the case of a mismatch, the value at the server is assumed to be correct. To help optimize performance, always specify the correct authentication type at the client to avoid this extra network flow

Authentication architecture



Upon db2start, db2ckpwd is spawned. Number of db2ckpwd base on DB2_NUM_CKPW_DAEMON. Default is 3.

Userid/pwd validation is done via db2ckpwd. OS calls are done via db2ckpwd



1. Upon connection, db2sysc spawns db2agent as thread
2. db2agent pass userid/pwd for validation to db2ckpwdIf userid/pwd validation is successful. Authorization is done via db2sysc
3. After validation, db2agent will proceed with group enumeration
4. If all is good, connection will proceed

Setting number of db2ckpwd

- DB2_NUM_CKPW_DAEMON controls the number of check password daemon
- Prior DB2 v9.5
  - Password authentication is done via process spawned by db2ckpwd
  - Set DB2_NUM_CKPW_DAEMON=0 is used to workaround many AIX related authentication issues. E.g. memory leak or cache problem
- After DB2 v9.5
  - Password authentication is done via db2ckpwd
  - DB2_NUM_CKPW_DAEMON=3:FORK == DB2_NUM_CKPW_DAEMON=0
  - [:FORK] = db2ckpwd spawn a separate daemon to do authentication. This separate daemon is terminated upon authentication.
  - Force AIX password authentication process out of db2ckpwd

Windows Authentication

- Domain/Local authentication
- Domain/Local group enumeration
- Extending Windows security to DB2

Windows Authentication

- Domain
  - PDC – Primary domain controller
    - Contain master copy of Security Access Manager (SAM) database
    - SAM contains info about which user can log onto the domain, their passwords, and their groups
    - Records members of the domain
    - Records other domains/ trusted domains
  - BDC – Backup domain controller
    - Holds a copy of SAM database from the PDC
    - Can authenticate users to the domain on behalf of the PDC
    - Backup for PDC in case of PDC fails

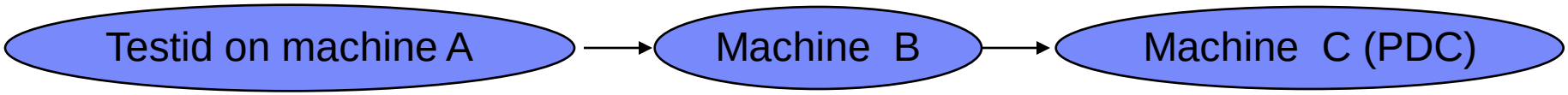
Windows Authentication - Groups

- Groups are maintained in the SAM database
- Two types of groups
  - Local group
    - Include user accounts have been created in the local accounts database
    - Can include local or domain accounts and groups
    - Local to the Windows machine
    - Store in the machine's SAM database
  - Global group
    - Define on domain controller
    - Contains only user accounts from domain's SAM
    - Known to the servers in its own domain and trusted domain
    - Can become a member of local groups of servers in its own domain and trusted domain
    - PDC holds the SAM for the domain. No local group for PDC. All user accounts/groups are for the domain

Windows Authentication of user – rule of thumb

- Order of looking for user
  1. Local machine
  2. Domain Controller of current domain
  3. Any trusted domains known to the domain controller
- After authentication is done – validated userid and pwd, DB2 proceeds with group enumeration

Windows – Server authentication in Single Domain



- Domain account testid on Machine A -> Machine B with userid/pwd
- Test is a domain account on PDC Machine C
- DB2 on Machine B searches local SAM
- If not found, search Domain Controller and determine testid is defined on Machine C
- If found to be a Domain account, DB2 on Machine B communicates with PDC to validate userid and password
- If password valid, continue to enumerate groups on Machine C
- Can change which group to enumerate base on DB2_GRP_LOOKUP

Windows Group Enumeration – rule of thumb

- Group enumeration
  - ALWAYS on the machine where user is defined
  - DB2 authenticate user in the order of
    - 1. look up user on local machine-> if user is defined locally, user group is enumerated locally
    - 2. If not found locally, look up user on its own domain controller -> if user is defined on domain controller, user group is enumerated on domain controller
    - 3. If not found on domain controller, look up user on trusted domain -> if user is defined on domain controller of trusted domain, user group is enumerated on domain controller of trusted domain
  - Group enumeration can override by DB2_GRP_LOOKUP setting

Windows – db2_grp_lookup

- By default - DB2 database manager enumerate groups on where the user is defined
  - Local user – local machine
  - Domain user – PDC of its own domain or trusted domain
- DB2_GRP_LOOKUP – control how DB2 perform group lookup on Windows OS
  - Consist of one or two parameters
    - First parameter: “ “, LOCAL, DOMAIN
      - Used the conventional NetApi32 (NetUserGetLocalGroup, etc) good for NT domains

First param	Enumerate at	Remarks
“ “	where user is defined Local user – local machine Domain user – at the domain	
Local	Local user – local machine Domain user – local machine	Local groups can contain local user, domain user, and global group
Domain	Deprecated same as “ “	

Windows – db2_grp_lookup - token

- Second parameter: TOKEN, TOKENDOMAIN, TOKENLOCAL
  - NetApi32 is too restrictive
  - Took advantage of token. If token group enumeration fails, will fallback to conventional NetApi32 group lookup
- Access token
  - object describes security context of a process or thread
  - Include identity and privileges of user account associated with the process/thread
  - include all groups the account belong to (local and domain)
  - Advantage of support nested global groups, domain local groups
  - When log on and password is authenticated, system produce access token for this user
  - Every process run on this account uses the copy of this access token

Windows – DB2_GRP_LOOKUP - token

Second param	Enumerate at
TOKENLOCAL	Same as conventional “LOCAL” Local user – local machine Domain user – local machine
TOKENDOMAIN	Same as conventional “ ” Local user – local machine Domain user – at the domain
TOKEN	Both local and domain (no equivalent from conventional group lookup) Local user – local machine only (local user cannot be part of global group) Domain user - both local and domain

Windows – DB2_GRP_LOOKUP - token

- Limitation of access token
  - Access token is used against the user connecting to database only. When another auth id needs to be group enumerated (e.g. under a different session user by set session_user), conventional group enumeration is used.
  - E.g. fenced SP is ran under fenced userid (not the connect userid), so access token cannot be used for fenced user
- Best practice:
  - Set db2_grp_lookup with conventional method and access token (BOTH)
  - Local, tokenlocal
    - access token – local groups
    - Different userid than connect userid – local group lookup at DB2 db
  - Token
    - Access token – where user is defined
    - Different userid than connect userid – where user is defined
  - Domain, tokendomain
    - Access token – domain groups
    - Different userid than connect userid – where user is defined

Windows – User start DB2 Service – Local account

- If DB2 users needs to be authenticated is local then authentication is done locally and group enumeration is done locally
- Userid that starts DB2 service needs rights to perform authentication of any user with below rights
  - Read Group Membership
  - Read groupMembershipSAM
  - Act as part of the operating system
  - Create a token object
  - Increase quotas
  - Log on as a service
  - Replace a process level token
  - Lock pages in memory

Windows –SYSADM_group

- When sysadm group is not defined in dbm cfg, then by default, the sysadm members are:
  1. Members in local Administrators group if user is local account
  2. Members in domain's administrators group if the user is domain account
  3. If extended security is enabled, members of DB2ADMNS group. DB2ADMNS (local or global) is defined when DB2 is installed

Windows – Setting SYSADM group

- To override sysadm_group:
  - Local sysadm group - Create a local group on DB2 server and set this local group as sysadm_group. Set group enumeration to be done on local machine
  - Global sysadm group - Create a domain group and set this group as sysadm_group. Set group enumeration to be done on where user is defined.
    - What if group enumeration is set to local, what will happen?
    - If Global group is used as SYSADM group, then you must:
      - 1. Include the global group in the local group of where DB2 is installed.
      - 2. Grant permission to this local group
      - OR
      - 1. Include the global group on the domain controller
      - 2. Grant permission to this global group
      - 3. Users in this group are global user accounts

To find out which group the user belongs to

- AUTH_LIST_GROUPS_FOR_AUTHID table function
- db2 get authorization
- OS commands:
  - net user <username>
  - net localgroup
  - net localgroup Administrators
  - net localgroup DB2ADMNS
  - net localgroup Users
  - regedit /e db2servlist.txt
  - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
  - Net user <username> /domain – user accounts from PDC of current domain
  - Net group /domain – global groups from PDC of current domain
  - Net group <global group> /domain

To find out which group the user belongs to

- AUTH_LIST_GROUPS_FOR_AUTHID table function
- SELECT * FROM TABLE (SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID (‘testid’)) AS T

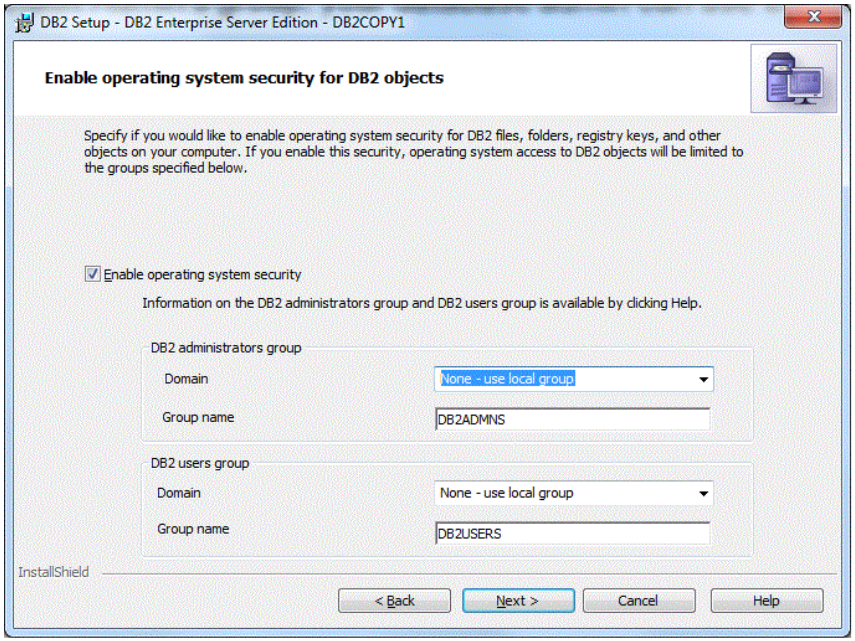
```
C:\Program Files\IBM\SQLLIB\cfg>db2 SELECT * FROM TABLE (SYSPROC.AUTH_LIST_GROUP
S_FOR_AUTHID (‘test')) AS T

GROUP

------------------------------------------------------------------------------
--------------------------------------------
DB2ADMNS

DB2USERS

ADMINISTRATORS
```

Extending Windows security to DB2

- Extended security is enabled by default in all DB2® database products on Windows operating systems except IBM Data Server Runtime Client and DB2 Drivers.
- You see a check box for this option upon DB2 install.
- This option is check by default and set to groups names of DB2ADMNS and DB2USERS.
- You can to rename the two the groups DB2ADMNS and DB2USERS.
- If you select silent install, you can change these names within the install response file.
- You may also choose to use groups that already exist on your system, be aware that the privileges of these groups will be modified.

Extending Windows security to DB2

- It is important to understand that these groups are used for protection at the *operating-system* level and are in no way associated with DB2 authority levels, such as SYSADM, SYSMAINT, and SYSCTRL.
- However, instead of using the default Admin's group, your database admin can use the DB2ADMNS group for one or all of the DB2 authority levels, at the discretion of the installer or admin.
- It is recommended that if you are specifying a SYSADM group, then that should be the DB2ADMNS group. This can be established during installation or subsequently, by an admin.



Extending Windows security to DB2

- **Abilities acquired through the DB2ADMNS and DB2USERS groups**
- The DB2ADMNS and DB2USERS groups provide members with the following abilities:
  - DB2ADMNS Full control over all DB2 objects (see the list of protected objects, below)
  - DB2USERS Read and Execute access for all DB2 objects located in the installation and instance directories, but no access to objects under the database system directory and limited access to IPC resources
    - For certain objects, there may be additional privileges available, as required (for example, write privileges, add or update file privileges, and so on). Members of this group have no access to objects under the database system directory.
    - Note: The meaning of Execute access depends on the object; for example, for a **.dll** or **.exe** file having Execute access means you have authority to execute the file, however, for a directory it means you have authority to traverse the directory.

Extending Windows security to DB2

- If you change the computer name, and the computer groups DB2ADMNS and DB2USERS are local computer groups, you must update the DB2_ADMINGROUP and DB2_USERSGROUP global registries. To update the registry variables after renaming and restarting the computer run the following command: Open a command prompt.
  - Run the db2extsec command to update security settings:
    - db2extsec -a *new computer name*\DB2ADMNS -u *new computer name*\DB2USERS

Extending Windows security to DB2

- **Removing extended security**
  - CAUTION:
    - Do not remove extended security after it has been enabled unless absolutely necessary.
    - You can remove extended security by running the command
      - **db2extsec -r**
    - However, this will only succeed if no other database operations (such as creating a database, creating a new instance, adding table spaces, and so on) have been performed after enabling extended security. The safest way to remove the extended security option is to uninstall the DB2 database system, delete all the relevant DB2 directories (including the database directories) and then reinstall the DB2 database system without extended security enabled.
- For more information on extending Windows security to DB2 please use the following link
  - DB2 and Windows security introduction

LDAP authentication

- LDAP plug in authentication
  - Supported starting v9.1 FP6
  - LDAP plug in configuration
    - IBMLDAPSecurity.ini setup
    - Enable plugin in dbm cfg
    - PD/PSI
      - Ldap commands, ldap trace, ldap plugin trace, samples
- Transparent LDAP authentication
  - Supported:
    - AIX, v9.1 fp 7, v9.5 fp 4 and v9.7 GA.
    - Linux, HP, and Sun V9.5 fp 5 and v9.7 fp 1

LDAP plug-in authentication

- Supported on OS:
  - AIX® Version 5.2, 5.3, and later
  - HP-UX on Itanium-based HP Integrity Series systems (IA-64)
  - Linux® distributions RHEL 4, SLES 9, and SLES 10 on x86, x86–64 and 64–bit zSeries® or System z9® hardware
  - Solaris 9 and Solaris 10
  - Windows® 2000, Windows 2003,Windows XP (on x86 and x86–64 hardware) Windows 2008, and Windows 7
- Supported LDAP products:
  - Supported LDAP servers for use with security plug-in modules are:
  - IBM® Lotus® Domino® LDAP Server, Version 7.0, and later
  - IBM Tivoli® Directory Server (ITDS) Version 5.2, 6.0, and later
  - Microsoft® Active Directory (MSAD) Version 2000, 2003, and later
  - Novell eDirectory, Version 8.7, and later
  - OpenLDAP server, Version 2.3.32, and later
  - Sun Java™ System Directory Server Enterprise Edition, Version 5.2, and later
  - z/OS® Integrated Security Services LDAP Server Version V1R6, and later

LDAP plug-in authentication

- When LDAP plug-in is used, all DB2 users (instance owner, fenced user, other DB2 users) must be defined on the LDAP server
- When LDAP group plug-in is used, all groups must be defined on the LDAP server
- Plug-in modules are available for:
  - Client plug-in
  - Server plug-in
  - Group plug-in
- To use DB2 security plug-in modules, follow these steps:
  1. Decide if you need server, client, or group plug-in modules, or a combination of these modules.
  2. Configure the plug-in modules by setting values in the IBM LDAP security plug-in configuration file (default name is IBMLDAPSecurity.ini). Consult with LDAP administrator to determine appropriate values.
  3. Enable the plug-in modules.
  4. Test connecting with various LDAP User IDs.

LDAP plug-in modules

Client plug-in	Server plug-in	Group plug-in
Validation of userid/pwd (supplied by client on CONNECT and ATTACH) occurs on client machine DB2 Server is configured with authentication=CLIENT or srvcon_auth= CLIENT	-validation of userid and pwd (supplied by client on CONNECT and ATTACH) occur on server Map LDAP user IDs to DB2 authorization IDs -Generally required if want users to authenticate to DB2 DBM using LDAP userid and pwd	-retrieve group information from LDAP server -LDAP server store all group definitions -Sample use: -All users and groups defined in ldap server -Any users defined locally on db server also defined with same userID on ldap server -Authentication or srvcon_auth set to SERVER - group info pull from LDAP

LDAP plug-in – configure plug-in modules

- Configure LDAP plug-in modules by updating IBM LDAP security plug-in config file:
  - Unix: INSTHOME/sqllib/cfg/IBMLDAPSecurity.ini
  - Windows: %DB2PATH%\cfg\IBMLDAPSecurity.ini
- Server values

Parameter	Description
LDAP_HOST	-LDAP server name/ IP addr. -List of ldap server host name/ IP addr. is separated by space Host1[:port1] host2[:port2] Default port is 389, SSL enabled is 636
ENABLE_SSL	True – enable SSL support Default to NO
SSL_KEYFILE	Path for SSL keyring Only required if LDAP server using certificate not automatically trusted by GSKit installation
SSL_PW	SSL keyring password

LDAP plug-in authentication

- User related values

Parameter	Description
User_objectclass	-LDAP object class used for users -Generally set to inetOrgPerson (user for MSAD)
User_BaseDN	-LDAP base DN to use when search for users -If not specified, search from root of LDAP directory (some ldap server require this set to something) USER_BASEDN= o=ibm
Userid_attribute	-LDAP user attribute represents userid -combined with user_objectclass and user_basedn to construct LDAP search filter -E.g. User_attribute=uid, User_attribute=sAMAccountName (for MSAD) -Connect to db user testuid using pwd -Search filter is &(objectClass=inetOrgPerson)(uid=testuid)
AuthID_Attribute	LDAP user attribute respresents DB2 authorization ID Usually same as Userid_Attribute

LDAP authentication – Enable LDAP plug-in

- Use the DB2 command line processor to update the database manager configuration to enable the plug-in modules that you require:
  - For the server plug-in module: `UPDATE DBM CFG USING SRVCON_PW_PLUGIN IBMLDAPauthserver`
  - For the client plug-in module: `UPDATE DBM CFG USING CLNT_PW_PLUGIN IBMLDAPauthclient`
  - For the group plug-in module: `UPDATE DBM CFG USING GROUP_PLUGIN IBMLDAPgroups`

Transparent LDAP

- When setting up transparent LDAP you need to use the below link and use the steps outlined for the OS you have DB2 installed on:
- LDAP-based authentication and group lookup support
  - **Collecting Data for DB2 LDAP Authentication**
  - Set the DB2AUTH miscellaneous registry variable to OSAUTHDB. As a user with SYSADM authority run db2set DB2AUTH=OSAUTHDB.
    - Using the UPDATE DBM CFG command, set the authentication on the database server instance to any one of the following:
      - SERVER
      - SERVER_ENCRYPT
      - DATA_ENCRYPT
  - Ensure that you are using the default Client Userid-Password Plugin (clnt_pw_plugin), Server Userid-Password Plugin (srvcon_pw_plugin) and Group Plugin (group_plugin).
  - Restart the DB2 instance.
- If the steps outlined in the above link are not followed exactly DB2 has been known to crash or hang due to not being able to communicate correctly with the LDAP server.

DB2 for z/os Basic terminology

- DB2 for z/os is a subsystem that runs in the z/os operating system
- There can be one or multiple DB2 subsystems on a z/os image
- DB2 for z/os can be either a standalone subsystem or can be a member of a DB2 data sharing group. A DB2 data sharing group is multiple DB2 subsystems coupled as a single group that share data. This is a typical configuration for high availability environments.
- A DB2 for z/os subsystem is made up of multiple DB2 databases. Each database contains multiple objects such as tables and indexes.
- When a client connects to DB2 for z/os, you connect to a DB2 subsystem or DB2 data sharing group.

What you need to know to connect to DB2 for Z/OS

- Information required from DB2 for Z/OS when setting up a connection for a remote client

- The –DISPLAY DDF command output on DB2 for z/os will provide the necessary information for setting up a connection.

```
DSNL080I - DSNLTDDF DISPLAY DDF REPORT FOLLOWS:
DSNL081I STATUS=STARTD
DSNL082I LOCATION          LUNAME        GENERICLU
DSNL083I STL717A           USIBMSY.SYEC717A    -NONE
DSNL084I TCPPORT=446    SECPORT=0      RESPORT=5001   IPNAME=-NONE
DSNL085I IPADDR=::9.30.115.135
DSNL086I SQL     DOMAIN=v7ec135.svl.ibm.com
```

* You will need to know the DB2 for z/os location name (DSNL082I) , the TCPPORT (DSNL084I) and either the ipaddr (DSNL085I) or the SQL domain name (DSNL086I) for the DB2 subsystem you want to connect to.

DB2 for Z/OS authentication and security

- DB2 for Z/OS calls IBM RACF or other OEM security software to authenticate the userid and password
- During the handshake of DB2 for z/os and the client, the type of security to be used is negotiated to a level and type that both requester and server support.
- Once authenticated to access z/os and the DB2 for z/os subsystem, authorization to objects within DB2 for z/os can be done in two ways:
  - DB2 can manage authorizations
  - External security product outside of DB2

Sending encrypted passwords from workstation clients

- As a server, DB2® for z/OS® can accept requests from remote workstation clients that use 256-bit Advanced Encryption Standard (AES) or 56-bit Data Encryption Standards (DES) encryption security over a TCP/IP network connection
- A remote client can use AES or DES encryption algorithm for sending passwords, user IDs and passwords, or other security-sensitive data to a DB2 for z/OS server. If both the DB2 for z/OS server and the remote client support DRDA Security Manager (SECMGR) 9 or higher and even if the client does not explicitly request for AES, AES becomes the default encryption algorithm for user IDs and passwords, and DES remains the default encryption algorithm for security-sensitive data. In other words, if the client explicitly requests for AES encryption, only user IDs, passwords, or both are encrypted in AES, and any data in the request is still encrypted in DES. Any persistent attempt to encrypt the data in AES will cause the client itself to reject the connection request.
- To use the DES encryption, you can enable DB2 Connect™ to send encrypted passwords by setting database connection services (DCS) authentication to DCS_ENCRYPT in the DCS directory entry. When a client application issues an SQL CONNECT, the client negotiates this support with the database server. If supported, a shared private key is generated by the client and server using the Diffie-Hellman public key technology and the password is encrypted using 56-bit DES with the shared private key. The encrypted password is non-replayable, and the shared private key is generated on every connection. If the server does not support password encryption, the application receives SQLCODE -30073.

Sending encrypted passwords from workstation clients … AES

- To enable the DB2 for z/OS AES server support, you must install and configure z/OS Integrated Cryptographic Services Facility (ICSF). During DB2 startup, DB2 invokes the MVS LOAD macro service to load various ICSF services, including the ICSF CSNESYE and CSNESYD modules that DB2 calls for processing AES encryption and decryption requests. If ICSF is not installed or if ICSF services are not available, DB2 will not be able to provide AES support. Instead, it will use DES for processing remote requests if the client does not explicitly request for AES encryption.

- *Additional information on ICSF including hardware and software requirements can be found in the ICSF administration guide*

Encrypted password, user ID, or user ID and password security under the IBM Data Server Driver for JDBC and SQLJ

- IBM® Data Server Driver for JDBC and SQLJ supports encrypted password security, encrypted user ID security, or encrypted user ID and encrypted password security for accessing data sources.
- The IBM Data Server Driver for JDBC and SQLJ supports 56-bit DES (weak) encryption or 256-bit AES (strong) encryption. AES encryption is available with IBM Data Server Driver for JDBC and SQLJ type 4 connectivity only. You set the encryptionAlgorithm driver property to choose between 56-bit DES encryption (encryptionAlgorithm value of 1) and 256-bit AES encryption (encryptionAlgorithm value of 2). 256-bit AES encryption is used for a connection only if the database server supports it and is configured to use it.
- If you use encrypted password security, encrypted user ID security, or encrypted user ID and encrypted password security from a DB2® for z/OS® client, the Java Cryptography Extension, IBMJCE for z/OS needs to be enabled on the client.
- The Java Cryptography Extension is part of the IBM Developer Kit for z/OS, Java 2 Technology Edition. For information on how to enable IBMJCE, go to this URL on the web: http://www.ibm.com/servers/eserver/zseries/software/java/j5jce.html
- For AES encryption, you need to get the unrestricted policy file for JCE. It is available at the following URL: https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk

DB2 for z/os Encrypted security-sensitive data under the IBM Data Server Driver for JDBC and SQLJ

- You can also use encrypted security-sensitive data in addition to encrypted user ID security to DB2 for Z/OS.
- You specify encryption of security-sensitive data and user through the ENCRYPTED_USER_AND_DATA_SECURITY

NOTE: This is method is valid for connections to DB2 for z/OS servers only , and only for DES encryption (encryptionAlgorithm value of 1).

- DB2 for z/OS database servers encrypt the following data when you specify encryption of security-sensitive data:

  - SQL statements that are being prepared, executed, or bound into a package
  - Input and output parameter information
  - Result sets
  - LOB data
  - XML data
  - Results of describe operations

NOTE: Before you can use encrypted security-sensitive data, the z/OS Integrated Cryptographic Services Facility needs to be installed and enabled on the z/OS operating system

Configuring the z/OS LDAP server and Setting up RACF for the z/OS LDAP server

When DB2® receives an authenticated user registry name, it invokes the SAF user mapping plug-in service. This service uses the EIM domain, which is an LDAP server, to retrieve the z/OS® user ID that is used as the primary authorization ID.

- **To configure a z/OS LDAP server refer to the steps at this link**
  - http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z9.doc.admin/src/tpc/db2z_configureldap4eim.htm

- **After you configure the z/OS® LDAP server, you need to set up RACF® to activate identity mapping. You also need to grant DB2® authority to use the SAF user mapping plug-in service**.
  - http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z9.doc.admin/src/tpc/db2z_configureracf4eim.htm

- **After you set up the LDAP server and RACF®, you need to use the RACF eimadmin utility to create and configure an EIM domain controller**.
  - http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z9.doc.admin/src/tpc/db2z_setupeimdomain.htm

- **Adding the SAF user mapping plug-in data set to LNKLIST**
  http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z9.doc.admin/src/tpc/db2z_addmapdataset2lnklist.htm

DB2 for z/os references

- DB2 10 for z/os info center home page

http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc/src/alltoc/db2z_10_prodhome.htm

DB2 9.1 for z/os info center home page

http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z9.doc/src/alltoc/db2z_09_prodhome.htm

- IBM REDBOOK

DB2 9 for Z/OS:Distritbuted Functions (sg24-6952)

Common DB2 LUW issues and references

SQL30082N Security processing failed with reason "19" ("USERID DISABLED or RESTRICTED"). SQLSTATE=08001	Information and how to resolve error SQL30082 rc= 19.
Error SQL30082N Reason Code 15 or 24	Information and how to resolve error SQL30082 rc=15 and rc=24.
DB2 Security, Part 11: Develop a security plug-in for DB2 database authentication	Information of how DB2 works with LDAP.
Use Technology Explorer for IBM DB2 to manage user and group authentication for DB2 for Linux, UNIX, and Windows	Information about different types of authentication that work with DB2.
DB2 UDB support for authentication of users via LDAP	Information about LDAP and Transparent LDAP authentication setup with DB2.
DB2 UDB security, Part 6: Configure Kerberos for authentication on DB2 UDB for Linux, UNIX, and Windows	Information about Kerberos working with DB2 and how to set it up.
Learning more about DB2 security model on Windows platforms	A collection of knowledge now on how Windows authentication works with DB2.

Team Acknowledgement

- Connie Lam
  DB2 LUW Security Team

- Gorge Makovich
  SWG Client Support – Software

- Mary Ann Ritosa
  DB2 for z/OS Technical Support

Adam Koile
DB2 LUW Security Team

Q&A