Database Management Group (DBM)

Enterprise Computing Services

DB2 Security, Access, and Performance for the Internet

POLICY/GUIDELINE NUMBER: 010212-01 **Approved By:** Edward J. Hively

SUBJECT: Security, Access, and Performance issues for DB2 Universal Database for z/OS.

Introduction: Securing and protecting our client's data has always been our primary concern. The need to provide access to information to legitimate parties, and to block access to information by unknown and unauthorized parties. The core conflicts have been access type and of controlling potential adversaries from abusing secured data.

Purpose/Scope: This policy has been established to protect the integrity of the State of Nebraska data from fraudulent use and/or abuse resulting from unauthorized access. The five fundamental objectives are (1) controlling access, (2) authentication, (3) accountability, (4) privacy, and (5) data integrity (see Appendix for definitions).

- Section I addresses the Internet type processing.
- Section II discusses additional security concerns.

Section I. INTERNET processing.

A. Issues/Concerns.

Internet access has become a new platform for distributing DB2 for z/OS data across the networks and computer systems linked together. With this openness brings many potential issues that have to be addressed by Management, Developers, and our Client's. We are now working outside our secure mainframe (i.e. host) environment and have to "think" security when we and implement Internet applications.

For those developing Internet applications here are a few issues to address.

- Generic interfaces (API's) such as ODBC (Open Database Connectivity) and JDBC (Java Database Connectivity) development add a new 'dynamic' complexity into security, access, and performance issues.
- ODBC/JDBC access requires dynamic SQL execution. Dynamic SQL requires a DB2 bind of all statements, which places locks on catalog and directory tables for the duration of the executed statements. This is an iterative process; therefore, it is repeated each time dynamic SQL is executed.
- Dynamic SQL requires the end users to have privileges to access the relational data. This is difficult for your DBMS to manage and control.

- Use of "system or surrogate" userid's to run the application causes many problems. You lose the ability to track the activities of individual users with the database with regard to security, auditing, and performance.

- Dynamic SQL cannot be controlled and, therefore we cannot guarantee that it happened.

- Dynamic SQL requires bind authority and execution authority to be given to the person who actually submits the dynamic SQL.

- Dynamic SQL places locks on both application DB2 objects (tablespaces, tables, and indexes) and system DB2 objects (system catalog tablespace, tables, and indexes).

- Dynamic SQL causes major performance problems over static SQL (static SQL is the preferred method of coding along with calling stored procedures).

- With Dynamic SQL you have accountability issues. There is no assurance that any transaction that takes place can be subsequently proven.

- SQL statements that are dynamic are not logged and cannot be deciphered through an audit. Therefore, no DB2 'undo' log record for recovery is possible.

- DRDA (Distributed Relational Database Architecture) connections directly to DB2 using TCP/IP have fewer security controls than do connections using SNA protocols. Therefore, access to our client's data is open for unauthorized access.

B. Resolution.

Having stated the Security, Access, and Performance Issues/Concerns in Section A, there are legitimate reasons for opening up the DB2 for Z/OS Enterprise system to Internet access.

Therefore, the following criteria must be met before we can allow this type of access.

- Any Web service will be required to connect to DB2 for z/OS via the Recoverable Resource Services Attachment Facility (RRSAF). This allows for tracking Primary and SQL ids in the DB2 for Z/OS subsystem. It assures that the information that arrives is the same as when it was sent.

    - *This requirement meets the Integrity objective.*

- Writing Java Servlet's (written for JDBC) on the z/OS platform will be the preferred method of accessing DB2 for Z/OS data. This allows for programming access and execution to be controlled by web team.

- Java Servlet's (written for JDBC) will be required to call DB2 for z/OS Stored Procedures (SP's). The SP's are static SQL, thereby providing a better tool for access, performance, and business logic (as opposed to dynamic SQL). This narrows the dynamic SQL, executed via servlet, to only accessing certain data.

    - *This requirement meets the Controlling Access security objective.*

- Java Servlet's (written for JDBC) will be required to execute the 'set current sqlid=' statement to pick up authority to execute certain Stored Procedures. This assures that the Id executing the servlet is permitted to do so.

    - *This requirement meets the Authentication objective.*

- ❖ Stored Procedures will be run under either CLI (type 1) drivers or ODBC (type 2) drivers to accessing DB2 data. This assures that sensitive information is limited and capable of being verified.

    - *This requirement meets the Privacy objective.*

- ❖ DRDA (Distributed Relational Database Architecture) connections that use TCP/IP will have access to DB2 data via the WebServer for z/OS Server using SSL (Secured Socket Layer) ports. No direct DRDA connections, using TCP/IP, to DB2 will be allowed. This assures that the sensitive information is not visible and that the computer at the other end of session is permitted to access this data.

    - *This requirement meets the Privacy and Controlling Access objectives.*

Section II. Addition Security Concerns.

To make our environment secure, the data provided in the DB2 for z/OS enterprise must be determine 'to be' of value and then apply the appropriate level of technical and procedural security. To wait for our data to be unavailable (i.e. through a denial of service attack) is unthinkable. Management has to decide how to protect the enterprise information from the different paths of access. Previously, the batch and CICS were the only forms of accessing this data. Today, with Distributed and Internet platforms access our concerns bring more challenges to the Database Management Systems (DBMS).

The main areas to address when considering going to the Internet are:

- Database/Application Security design considerations.
- User Id and passwords to access this data.
- Access control built into the application.
- User authentication to the database.
- Encrypting the information in the database when there is a need for confidentiality.
- Separate Security Administration should be considered.
- Third Party Database Security Products should be features built in to the design.

Bottom Line: As we explore more avenues to provide our customers with access to their data, the enterprise must implement security controls in the end-to-end web to database environments. This means we must improve security coverage by access, authentication, authorization, confidentiality, and privacy controls

Section III. Appendix.

A. Definitions.

(1) Controlling Access – Assurance that the person (or computer) at the other end of the session are permitted to do 'what they ask for'.

(2) Authentication – Assurance that the resource (person or computer) at the other end of the session 'really' is what it claims to be.

(3) Accountability – Assurance that any transaction that takes place can subsequently be proven to have taken place. Both the 'sender' and the 'receiver' agree that the exchange took place.

(4) Privacy – Assurance that sensitive information is not visible to any eavesdropper.

(5) Integrity – Assurance that the information that arrives is the same as when it was sent.