DB2 Security Overview

Deb Jenson
Product Manager, Data Studio
dejenson@us.ibm.com
Disclaimer

This presentation is intended to provide general background information, not regulatory, legal or other advice. IBM cannot and does not provide such advice. Readers are advised to seek competent assistance from qualified professionals in the applicable jurisdictions for the types of services needed, including regulatory, legal or other advice.
Agenda

- The Importance of Data Security
- IBM Data Server Security Blueprint
- Key Security Features found in DB2
  - Authentication
  - Authorization
  - Database Roles
  - Trusted Contexts
  - Label-Based Access Control
  - Auditing
  - Encryption
  - Static SQL / pureQuery
  - Test Data Privacy

- Summary
The Importance of Data Security
The Importance of Data Security

- Historically focus on physical, network, and host security
- But database is where the valuables are kept!
- Data security has now moved to forefront, mostly due to rash of large breaches

Source: Flowingdata.com
Rogue DBA Steals, Sells Personal Info

Jaikumar Vijayan

Today’s Top Stories or Other Security Stories

Comments (0)  Recommendations: 72 — Recommend this article

July 09, 2007 (Computerworld) — Call it a case of hiring a fox to guard the henhouse.

Fidelity National Information Services Inc. said last week that a senior database administrator responsible for defining and enforcing data access rights at one of its subsidiaries sold the personal information of about 2.3 million consumers to a data broker. The broker in turn sold a subset of the data to “a limited number” of direct marketing companies, Fidelity National said.

The Jacksonville, Fla.-based company, which offers data processing and outsourcing services to financial institutions and other businesses, added that the stolen data included names, addresses, birth dates, and bank account and credit card information.

For now, at least, it appears that the companies that bought the information have used it mainly to send marketing solicitations to the affected individuals, according to Fidelity National. “We have no reason to believe that the theft resulted in any subsequent fraudulent activity,” said Renz Nichols, president of the company’s Certegy Check Services Inc. unit.

The database administrator has since been fired, and Fidelity National has filed a civil complaint in a court in St. Petersburg, Fla., against him and the companies that received the stolen data, seeking its return. Fidelity National said it also is “encouraging immediate prosecution” of the DBA by law enforcement.

MORE RELATED CONTENT

- Washington state works out $1M settlement with ‘safe surf’ vendor
- Firms found in breach of UK law on customer data
- House’s second antisympware bill rolls onward to the Senate

TODAY’S TOP STORIES

- How to track down network tie-ups
- Cisco co-founder launching optical network start-up
- Profile: Hayden Hamilton
- FastSoft speeds up WAN with FastTCP technology
- BT customers begin migration to IP network

IDG RELATED CONTENT

Managing encrypted data doesn’t have to be frustrating.

Simplify with centralized key management from Sun StorageTek.

Read the IDC White Paper

© 2008 IBM Corporation
Who Steals the Data?

- Hackers
  - TJ Maxx
    - >90 million credit and debit card numbers stolen
    - Estimated costs $1bn over five years ($117m costs in 2Q’07 alone)
- Insiders
  - Coca-Cola
    - Secretary gained access to a new recipe and tried to sell it to Pepsi
- Physical Thieves
  - Accenture
    - Unencrypted tape stolen
    - 58 taxpayers and 460 bank accounts
    - Litigation ongoing
Regulatory Compliance

- Many regulations exist today that mandate good practices
- Applicability depends on industry and country
- Some of the major governance regulations are:
  - PCI
  - Sarbanes-Oxley (SOX)
  - HIPAA
  - Data Breach Disclosure Laws
  - Gramm-Leach-Bliley
  - Basel II
Why Aren't all Databases Secure?

- Performance, features and scalability are often the requirements - security is usually an afterthought

- Performance vs. Security - guess who wins?

- DBA’s are not usually security people, and vise versa

- Lack of understanding of the threats

- Vendors usually present security as feature / function without context or what it protects against

- Security is taken care of in other layers – so why worry?
IBM Data Server Security Blueprint

- A Blueprint for effective data security
- Easy to use, single page
- Version 1.0.0 released March 2008
- DB2 for LUW, DB2 for z/OS and IDS
- Accompanying whitepaper
IBM Data Server Security Blueprint

Data Security
- Data Threats
- Configuration Threats
- Audit Threats
- Executable Threats

Host Security

Network Security

Physical Security

Identity Management

Business Controls
Counter Measures: Secure Data Server Features

- Ties DB2’s extensive security features to the threats they help protect against
  - Authentication
  - Authorization
  - Database Roles
  - Trusted Contexts
  - Label-Based Access Control
  - Auditing
  - Encryption
  - Static SQL / pureQuery
Using DB2 to Secure DB2

Deb Jenson
Product Line Manager
Data Studio
Key Security Features in DB2

Authentication
- Authorization
- Database Roles
- Trusted Contexts
- Label-Based Access Control
- Auditing
- Encryption
- Static SQL / pureQuery
Authentication DB2 LUW

- The process by which a user is validated
  - Performed outside of DB2 via authentication security plug-in
  - Default is OS based authentication (Shipped with DB2)

- Authentication Types
  - SERVER
  - SERVER_ENCRYPT
    - User ID and password encrypted
  - DATA_ENCRYPT
    - Data and User ID and password encrypted
  - KERBEROS
  - GSSPLUGIN
  - LDAP
  - CLIENT
Key Security Features in DB2

- Authentication
  
  Authorization
  
  - Database Roles
  - Trusted Contexts
  - Label-Based Access Control
  - Auditing
  - Encryption
  - Static SQL / pureQuery
Authorization

- The process of checking whether a user is allowed to execute a statement or command

- Involves granting a set of permissions available to the authorization id.

- Permissions can be obtained from 3 sources:
  - Permissions held by the authorization id itself
  - Permissions held by the authorization id’s groups and roles
  - Permissions held by PUBLIC
Authorization (Continued)

- Permissions are divided into authorities and privileges

- Authorities
  - System level authorities (e.g. SYSADM, SYSCTRL)
  - Database level authorities (e.g. DBADM, SECADM)

-Privileges
  - Required to perform specific actions on database objects
  - Database, Table, View, Indexes, Schema etc.
Authentication vs. Authorization

Authentication

Is this correct password for Mary?

Authorization

Does Mary have an authorization to perform SELECT FROM mytable?

CONNECT TO sample USER mary Using pwd

MYTABLE

SELECT * FROM mytable
Key Security Features in DB2

- Authentication
- Authorization

Database Roles
- Trusted Contexts
- Label-Based Access Control
- Auditing
- Encryption
- Static SQL / pureQuery
Database Roles

- **What is a database role?**
  - A database object that groups together one or more privileges, authorities, security labels or exemptions
  - Can be granted to users, groups, PUBLIC or other roles

- **What is the advantage of database roles?**
  - Simplify the management of privileges in a database
  - SECADMs control access to their databases using the structure of their organizations
  - Control of specific roles can be delegated to others

Available in DB2 for LUW v9 and DB2 z/OS v9
Database Roles ( Continued )

- Assign users to roles not groups
  - Roles are controlled inside the database
  - Group privileges and authorities are not considered by DB2 when creating views, triggers, MQTs, static SQL and SQL routines
  - DB2 cannot know when membership in groups change so that it can invalidate the database objects (e.g., views) created by users who relied on their group privileges to succeed
Database Roles - Example

- **Usage scenario**
  - BOB and ALICE are members of the DEV department and have the privilege to SELECT from tables SERVER, CLIENT and TOOLS.

    - One day, management decides to move them to the QA department and the administrator has to revoke their privilege to select on tables SERVER, CLIENT and TOOLS.

    - Department DEV hires a new employee, TOM, and the administrator has to grant SELECT privilege on tables SERVER, CLIENT and TOOLS to TOM.
Database Roles – Example (Continued)

- **Without database roles:**

  ```sql
  GRANT SELECT ON TABLE SERVER TO USER BOB, USER ALICE
  GRANT SELECT ON TABLE CLIENT TO USER BOB, USER ALICE
  GRANT SELECT ON TABLE TOOLS TO USER BOB, USER ALICE

  REVOKE SELECT ON TABLE SERVER FROM USER BOB, USER ALICE
  REVOKE SELECT ON TABLE CLIENT FROM USER BOB, USER ALICE
  REVOKE SELECT ON TABLE TOOLS FROM USER BOB, USER ALICE

  GRANT SELECT ON TABLE SERVER TO USER TOM
  GRANT SELECT ON TABLE CLIENT TO USER TOM
  GRANT SELECT ON TABLE TOOLS TO USER TOM
  ```
Database Roles – Example (Continued)

- **With database roles:**

  CREATE ROLE developer

  GRANT SELECT ON TABLE SERVER TO ROLE developer
  GRANT SELECT ON TABLE CLIENT TO ROLE developer
  GRANT SELECT ON TABLE TOOLS TO ROLE developer

  GRANT ROLE developer TO USER BOB, USER ALICE

  REVOKE ROLE developer FROM USER BOB, USER ALICE

  GRANT ROLE developer TO USER TOM
Key Security Features in DB2

- Authentication
- Authorization
- Database Roles
- Trusted Contexts
- Label-Based Access Control
- Auditing
- Encryption
- Static SQL / pureQuery
Trusted Context

Typical Scenario
- Middle-tier authenticates users
- Single application userid is presented to the data server

Security and Authorization
- All data server privileges are granted to the application userid

Implications
- All users have the same level of authorization
- No preservation of original userid for auditing

Available in DB2 for LUW v9.5 and DB2 z/OS v9
Trusted Context

- Helps solves two important security challenges

1. Application servers use of a single user id
   - Loss of end user identity within the database server
   - Diminished user accountability
   - Over granting of privileges to a single authorization id

2. Lack of control on when privileges are applied
   - Lack of control of when privilege can be applied can weaken overall security
   - Today all connections are treated the same

Available in DB2 for LUW v9.5 and DB2 z/OS v9
What is a Trusted Context?

- A trust relationship between the database and an external entity such as an application server
- Stored in the database
- The trust relationship is currently based on the following trust attributes:
  - Authorization id
  - IP address (or domain name)
  - Data stream encryption
Trusted Context Example

CREATE TRUSTED CONTEXT appServer
BASED UPON CONNECTION USING SYSTEM AUTHID appServerID
ATTRIBUTES (ADDRESS ‘host-name1.dept.organization.com’,
ADDRESS ‘host-name2.dept.organization.com’
ENCRYPTION ‘HIGH’)

DEFAULT ROLE appServerRole
WITH USE FOR PUBLIC WITHOUT AUTHENTICATION,
Alice WITH AUTHENTICATION ROLE mgrRole

ENABLE
Advantages of Trusted Contexts

- **User accountability and compliance**
  - Eliminates shared user id with each person audited and accountable with their own user id
  - Allows switching of user id without having to rip down connection

- **Improved security**
  - More control on when privileges are available to users
  - Helps alleviate concern of misusing the system authid credentials to access the DB2 server
  - Enforce the least privilege security principle
Key Security Features in DB2

- Authentication
- Authorization
- Database Roles
- Trusted Contexts

Label-Based Access Control
- Auditing
- Encryption
- Static SQL / pureQuery
Label Based Access Control (LBAC)

- A flexible implementation of Mandatory Access Control (MAC)
- A security label is associated with both users and data objects

**Bob U Results**

<table>
<thead>
<tr>
<th>SERIAL_NUMBER</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN00000001</td>
<td>PO-HARRY-1</td>
</tr>
<tr>
<td>SN00000002</td>
<td>PO-TONY-1</td>
</tr>
<tr>
<td>SN00000005</td>
<td>PO-BRIAN-1</td>
</tr>
</tbody>
</table>

**Jane S Results**

<table>
<thead>
<tr>
<th>SERIAL_NUMBER</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN00000001</td>
<td>PO-HARRY-1</td>
</tr>
<tr>
<td>SN00000002</td>
<td>PO-TONY-1</td>
</tr>
<tr>
<td>SN00000003</td>
<td>Instructions</td>
</tr>
<tr>
<td>SN00000004</td>
<td>Release Notes</td>
</tr>
<tr>
<td>SN00000005</td>
<td>PO-BRIAN-1</td>
</tr>
</tbody>
</table>

**Artifact Table (protected table)**

<table>
<thead>
<tr>
<th>SERIAL_NUMBER</th>
<th>SECLABEL</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN00000001</td>
<td>Unclassified</td>
<td>PO-HARRY-1</td>
</tr>
<tr>
<td>SN00000002</td>
<td>Unclassified</td>
<td>PO-TONY-1</td>
</tr>
<tr>
<td>SN00000003</td>
<td>Confidential</td>
<td>Instructions</td>
</tr>
<tr>
<td>SN00000004</td>
<td>Secret</td>
<td>Release Notes</td>
</tr>
<tr>
<td>SN00000005</td>
<td>Unclassified</td>
<td>PO-BRIAN-1</td>
</tr>
<tr>
<td>SN00000006</td>
<td>Top Secret</td>
<td>MARS-35</td>
</tr>
</tbody>
</table>

```sql
select serial_number, title
from artifact
```

Available in DB2 for LUW v9 and DB2 z/OS v8
What Does LBAC Add to Table Protection?

- Does not replace the traditional discretionary access control – instead complements it at the row and/or column level

- The content of a table appears different depending on the user accessing that table

- No user has any inherent privileges to access the content of LBAC protected data even if they are DBADM!
LBAC is a Flexible MAC Implementation

- A security label is not a fixed structure
  - More then just “level” and “compartments”

- Security label structure is specified by the user

- Security administrators can protect different tables with different security policies within the same database

- LBAC can be used with roles, and trusted context features of DB2 together to provide
LBAC Label Components

Set

Array

Tree
Is LBAC Applicable in My Environment?

- LBAC was designed to address the needs of government, military and other regulated environments

- LBAC can also be used in non-regulated spaces if:
  1. One is willing and able to classify the data in a table
  2. Requirements can be mapped to a security label hierarchy
  3. Security requirements and corresponding hierarchy change infrequently within an organization
Key Security Features in DB2

- Authentication
- Authorization
- Database Roles
- Trusted Contexts
- Label-Based Access Control

Auditing
- Encryption
- Static SQL / pureQuery
DB2 Audit Facility for DB2 LUW 9.5

- Very configurable, low overhead

- **Audit Policy**
  - A database object that specifies what categories of events are to be audited

- **An audit policy can be applied to:**
  - A database
  - A table
  - A trusted context
  - An authorization id (user, role, group)
  - An authority (SYSADM, SYSCTRL, DBADM, SECADM etc.)
DB2 Audit Facility for DB2 LUW 9.5

What audit Categories can be specified?

- AUDIT – any access or configuration of the auditing system
- CHECKING – any authorization checks done by DB2
- CONTEXT – the big picture, lots of miscellaneous events
- EXECUTE – execution of SQL statements
- OBJMAINT – create/drop of objects, some alter
- SECMAINT – grant/revoke
- SYSADMIN – actions only SYSADM/DBADM can do
- VALIDATE - authentication
DB2 Audit Facility for DB2 LUW 9.5

- The DB2 instance and database can be audited independently
- Each database contains one or more audit policies that describe all the database operations that require auditing.
- The audit statement indicates which database objects are associated to those audit policies.
- The DB2 instance and database can be audited independently
- Each database contains one or more audit policies that describe all the database operations that require auditing.
- The audit statement indicates which database objects are associated to those audit policies.
- Once the database audit is activated the audit records are generated and stored in the database audit log file.
- Archiving creates the audit archive files, which contain time-sequenced audit records.
Key Security Features in DB2

- Authentication
- Authorization
- Database Roles
- Trusted Contexts
- Label-Based Access Control
- Auditing
- Encryption
  - Static SQL / pureQuery
## Protection from Theft

<table>
<thead>
<tr>
<th>Date</th>
<th>Location</th>
<th>Incident Description</th>
<th>Affected Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov. 6, 2008</td>
<td>Harvard Law School (Cambridge, MA)</td>
<td>A computer tape containing Social Security numbers, addresses, and financial information was either lost or stolen. About 8,000 records of present and former clients contained Social Security numbers; another 13,000 had other identification information that was contained on the tape.</td>
<td>21,000</td>
</tr>
<tr>
<td>Nov. 7, 2008</td>
<td>Christus Health Care (Houston, TX) (800) 877-9056</td>
<td>Two computer back-up tapes were stolen. Someone broke into a car in a Houston parking lot and took the tapes. The information on the tapes included patient names, Social Security numbers, demographic information, and in some cases, diagnosis codes.</td>
<td>Unknown</td>
</tr>
<tr>
<td>Nov. 7, 2008</td>
<td>Arizona’s Department of Economic Security (Phoenix, AZ)</td>
<td>(DES) is notifying the families of about 40,000 children that their personal data may have been compromised following the theft of several hard drives from a commercial storage facility. The information stored on the stolen disks included the names, addresses and phone numbers of families whose children were referred to the DES for early intervention services over the past several years. In the cases of families that had applied for and received services from the agency, their records also included Social Security numbers.</td>
<td>40,000</td>
</tr>
</tbody>
</table>

[www.privacyrights.org](http://www.privacyrights.org)
Encryption

- Used to ensure data privacy for sensitive data

- Used in to main areas:

1. Data in Transit
   - DATA_ENCRYPT Option
   - Secure Socket Layer (SSL)

2. Data at Rest
   - Column level encryption
   - File level encryption
Encryption – Data in Transit

- Data in Transit

  - DATA_ENCRYPT Option
    - Very simple to set - AUTHENTICATION configuration parameter
    - Automatically encrypts user data during client to server communications as it travels over the network
    - Available since version V8.2

  - Secure Socket Layer (SSL)
    - Stronger cryptographic algorithm
    - IBM DB2 Driver for JDBC and SQLJ type 4 connectivity supported
    - Recommended Option
Encryption – Data at Rest

Data at Rest

- **Column Level Encryption**
  - ENCRYPT / DECRYPT SQL Functions
  - Uses DES
  - Available since V7.2

- **File Level Encryption**
  - Increased Performance
  - Transparent to Application
  - Comprehensive protection (encrypt audit logs, txn logs etc.)
  - IBM Database Encryption Expert
  - Recommended Option
Encryption - Performance Benefits of File Level

- **Column Level Encryption is Slower**
  - Column level encryption adds significant overhead (40% – 50% or more), not due to cryptography but overhead of external calling mechanism
  - Index effectiveness is diminished particularly for range queries
  - Performance benefits of caching data rows is drastically reduced since the rows must first be decrypted

- **File level encryption is Faster**
  - Encrypts and decrypts data as DB2 reads or writes to database files which has no effects on data rows caching or indexing
  - Using IBM Database Encryption Expert, in-house testing with DB2 adds an overhead roughly 5%
Key Security Features in DB2

- Authentication
- Authorization
- Database Roles
- Trusted Contexts
- Label-Based Access Control
- Auditing
- Encryption

Static SQL / pureQuery
What is Static SQL?

- All SQL known before execution
  - Application and DBMS have both prepared for handling the SQL ahead of time

- SQL is parsed, optimized and bound into a “package” during build/deployment

- Gives database an application perspective of the running programs.

Available in DB2 for LUW and DB2 z/OS
### SQL Execution – Dynamic vs. Static

#### Dynamic SQL
1. Check auth for package / plan
2. Parse SQL Statement
3. Check Table / View Auth
4. Calculate access path
5. Store access path in a temporary package
6. Execute SQL statement

#### Static SQL
1. Check auth for package / plan
2. Extract access path from SYSCAT.PACKAGES and STATEMENTS
3. Execute SQL statements
# Static SQL: Major Advantages

<table>
<thead>
<tr>
<th>Feature</th>
<th>Dynamic SQL</th>
<th>Static SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Security</strong></td>
<td>• Privileges handled at object level.</td>
<td>• Privileges are package based.</td>
</tr>
<tr>
<td></td>
<td>• All users or groups must have direct table privileges.</td>
<td>• Only administrator needs direct table access. Users only need execute privilege.</td>
</tr>
<tr>
<td></td>
<td>• Security exposure, and administrative burden</td>
<td>• Prevent non-authorized SQL execution.</td>
</tr>
<tr>
<td><strong>Predictable Performance</strong></td>
<td>• Can approach static SQL performance but needs help from dynamic SQL cache</td>
<td>• All SQL parsing, catalog access, done at BIND time.</td>
</tr>
<tr>
<td></td>
<td>• Cache misses costly</td>
<td>• Fully optimized during execution.</td>
</tr>
<tr>
<td><strong>Access path Reliability</strong></td>
<td>• Unpredictable – Any prepare can get a new access path as statistics or host variables change</td>
<td>• Guaranteed – locked in at BIND time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• All SQL available ahead of time for analysis by EXPLAIN.</td>
</tr>
</tbody>
</table>
Static SQL : Secure Table Privilege Example

**Dynamic SQL**

Table privileges must be granted directly to the user, groups or role.

```
GRANT SELECT ON TABLE PAYROLL TO ROLE HR;
```

**Static SQL**

Users require no specific table privileges

```
GRANT SELECT ON TABLE PAYROLL TO ROLE BIND_ADM;
```

```
GRANT EXECUTE ON PACKAGE PAY_PKG TO ROLE HR;
```

**User**

**Database Admin**
pureQuery Give You The Best of Both Worlds

- pureQuery is a high-performance data access platform that simplifies the development and management of Java apps
- pureQuery APIs let you build a Java application with much less code vs. using JDBC or SQLJ
- Supports both Static SQL and Dynamic SQL
- Allows you to code to dynamic SQL, and turn on static SQL at deployment time!

Diagram:

- JDBC
- pureQuery
- SQLJ

Dynamic SQL
Static SQL

Runtime Control
Summary

- Data security has become critically important due to the increase of severe data breaches and regulatory compliance requirements.

- DB2 provides one of the industry's most secure data server environments.

- The IBM Data Server Security Blueprint get you started and helps simplify the task of securing your data server.
  - Threat oriented
  - Includes current recommendations of DB2 security team
  - Version 1.0.0 was released on March 2008
Summary

- The blueprint leverages DB2’s extensive security features. Key security features include:
  - Authentication
  - Authorization
  - Database Roles
  - Trusted Contexts
  - Label-Based Access Control
  - Auditing
  - Encryption
  - Static SQL / pureQuery

- The IBM Data Security Blueprint and accompanying whitepaper can be downloaded from:
  