



Security is in the headlines and growing much more important. This session will discuss various practices for security and discuss how you can make improvements. The discussion includes various security objectives. Most sites have a range of needs and objectives. For some situations, basic security is adequate. For others better or standard security techniques are needed. In other cases, best security practices are demanded. Our tools range from very tight system security to basic techniques, applicable with public information on the web. Application security techniques are more flexible, but require much more work by more people, so they are generally weaker. Choices and guidelines will be our primary points, discussing how to provide improved security for your situation.

For other details on DB2 for z/OS security, there are many suggested resources, such as the Protect Your Assets presentation:

<http://www.ibm.com/support/docview.wss?rs=64&uid=swg27001877>

Library for security:

<http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2.doc.admin/bjndmstr124.htm>

## Agenda

1. Security policy and objectives
2. Range of techniques
3. Practices: Basic, Standard, Best
4. Best practices and improvements
5. Checking and auditing

Get latest copy of presentation from web

<http://www.ibm.com/software/data/db2/zos/support.html>

This is the agenda for the presentation, beginning with a discussion of possible objectives, then looking through guidelines and techniques, while distinguishing the best practices from other security techniques. We will discuss ways to improve situations that I have seen before and emphasize the need for checking and auditing.

For more detail on DB2 security, see

<ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/db2sec1001p.pdf>

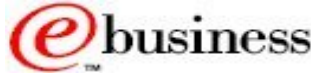
This presentation will be updated and placed on the DB2 web site. Check for it before and after conference time to get additions, corrections and many more notes.

<http://www.ibm.com/software/data/db2/zos/support.html>

Then click on Technical Presentations and put security into the additional search terms, sort results by date – newest first.

## Highly varied needs for

- ✓ Security
- ✓ Flexibility
- ✓ Ease of safe use
- ✓ Integration
- ✓ Assurance



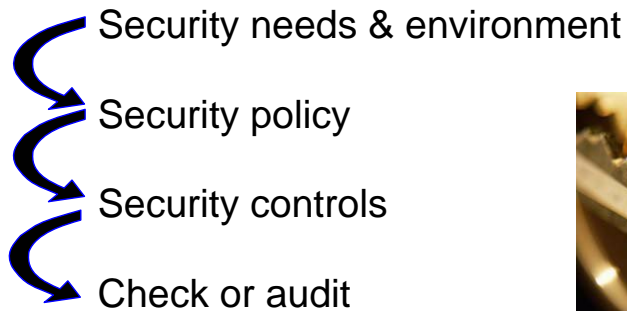
Everyone seems to be more aware of security today. Improving integration and making security more robust and easier to manage are very important.

Customers asked for a wide range of enhancements for security. New options for tighter security, more granularity, and more information for additional flexibility in applications and SQL are needed. Improved integration with the hardware and software platform provide a better end-to-end solution.

The design point for DB2 security is being easy to use for safe security, with solid system security and some security that is more flexible, but generally less secure, that we'll call application security.

Finally, it will be helpful to see what assurance can be provided, such as Common Criteria certification.

## Security Objectives & Needs



I'll try to go quickly through the introduction to security. I think this is important to make sure we start from the same point.

The needs for security are diverse – in several dimensions. The degree of security needed and the specific concerns differ. The needs and the environment should drive the security policy, which, in turn, drives the security controls. The objective is to match the controls as closely as possible with the security policy, filling in the details.

## What are your concerns? Level of concern?

Destruction of data

Changing data

Reading, confidentiality, disclosure or privacy

Denial of service

Audit trail

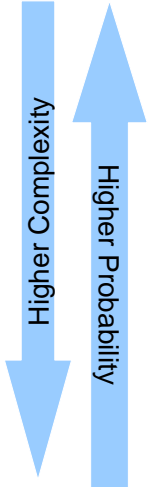
Here are some of the diverse dimensions and concerns for security. The needs will be driven by the type of business, the kinds of data and the business and legal issues for your enterprise. Financial information, health care data and defense have very different statements of security objectives, but some of the same principles apply. More commercial concerns for read access are arising from the need for privacy.

Still, the key concern for most commercial enterprises is being able to do the business, rather than the confidentiality of the information.

IBM Software Group | Information Management Software

IBM

## What kind of attacks do we face?



- Errors and Omissions
- Lost backups, in transit
- Application user (e.g. SQL injection)
- SQL users
- Network (e.g. LAN sniffer)
- Valid user for the server (e.g. stack overflow, data sets)
- Application developer, valid user for data
- Administrator

Page 6

See the next few slides and the operational environment slide for more potential of places which need to be addressed, including application code, web servers, database servers, directory and authentication devices, firewalls, network and enclave configuration and operating system platforms. It's important to understand the other security techniques and the controls to be sure there are no gaps in the fences. In general, we find more business losses from errors and omissions than from any other category. This area is a gateway to bigger problems, and one that can have a very positive return on investment.

## Security principles

Layered security

Individual accountability

Group & role authorization & ownership

Least privilege

Ease of safe use

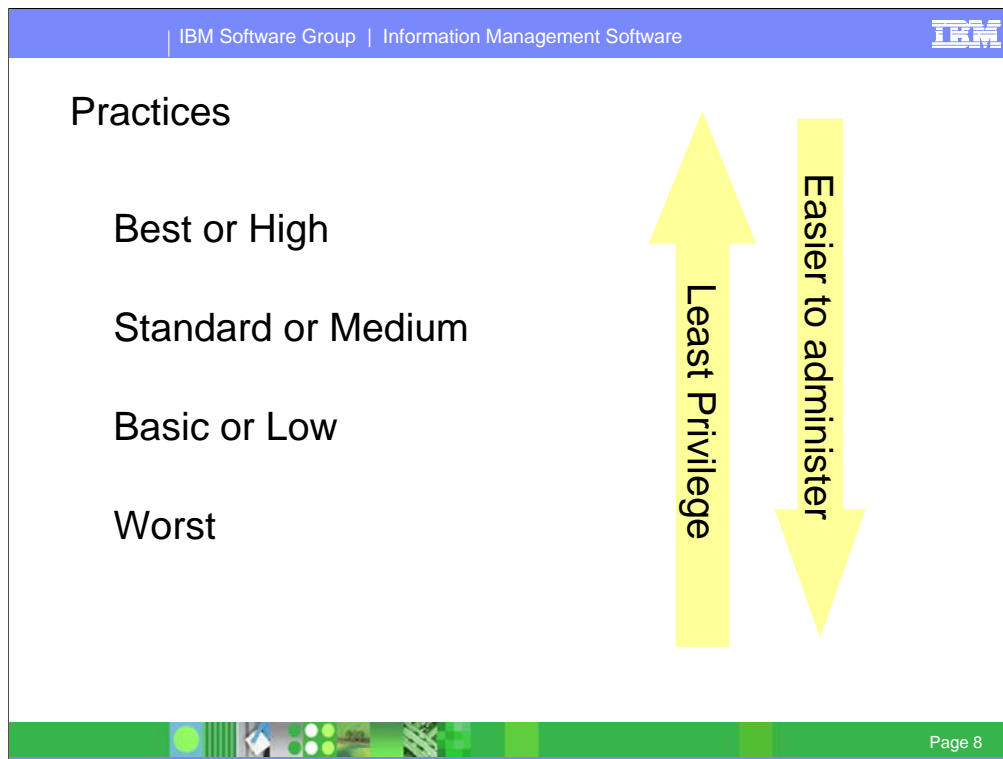


Since we know that systems are built and used in layers, the security also needs to be implemented in each layer.

Whenever possible, the authorization and ownership should be to or by a group or a role, but the individual who requests information needs to be understood to provide accountability.

“Least Privilege – This principle requires that each subject in a system be granted the most restrictive set of privileges (or lowest clearance) needed for the performance of authorized tasks. The application of this principle limits the damage that can result from accident, error, or unauthorized use.”

One of the primary concepts in security is giving the individuals only the privileges needed to do the job. That often means using more granular controls. Another key principle is ease of safe use. We want individuals to have all of the privileges they need to do the full job, but no more. If there is less complexity in the security controls, that means less cost and generally results in better compliance.



Security needs, dimensions and concerns for security vary widely. Some of the information needs best security practices, while other information can use standard or basic security to reduce the administration.

**Best** practice or high security will prohibit user access to a wide range of resources and require intensive access level checking. This mode may disable some less secure function and might reduce system performance. Access control is strict, with stringent audit and comprehensive protection.

**Standard** practice or medium security provides a balance of security with administration cost and system performance. The access control is moderately strict, with selective auditing. This level provides protection of resources that will be adequate for much of the information.

**Basic** practice or low security provides basic or minimal protection, with few constraints.. Minimal auditing is done. Protection is provided for some resources and a higher degree of risk is accepted for the lower cost of administration.

Worst practice provides essentially no security. One example is GRANT SYSADM TO PUBLIC or to groups that are broadly permitted, such as all data base administrators. Another example would be no use of audit.



## Layered security

Application code
Web servers
Database servers
Directory and authentication devices
Firewalls
Network and enclave configuration
Operating systems and platforms

Since we know that systems are built and used in layers, the security also needs to be implemented in each layer. The operating systems and platforms are the foundation for security. If the foundation is not secure and strong, then the other layers cannot protect themselves. If security is not stringent, then access to the operating system facilities must be restricted. The network can be the focus for an attack and for defense. Data on a LAN is often very vulnerable unless encryption is used. Firewalls can prevent some problems, but do not address many others. Database servers provide a wide variety of protection mechanisms. The lower levels have stronger defenses. If application code uses the security mechanisms, then the need for assurance is much less. If the application implements security, then much stronger assurance is required. If the application does not pass through the security information, then the ability to use database and operating system security can be compromised.

IBM Software Group | Information Management Software

IBM

## Primary security tasks

- identification & authentication
- access control
- confidentiality and privacy
- data integrity new redbook SG24-7111
- non-repudiation
- audit
- security management
- intrusion detection

IBM Information Management Software  
Draft document for Review May 26, 2009 10:40 pm  
**Data Integrity with DB2 for z/OS**  
IBM  
SG24-7111-00

Assert information integrity by exploiting DB2 functions  
Understand constraints, referential integrity and triggers

Page 10

Security control components are often categorized: Identification and authentication determine who the user is. Access control uses that identification to determine what resources can be used. Confidentiality and privacy controls help ensure that the access is allowable and monitored.

Data integrity controls are the basis for every database management system, with the ACID properties (atomicity, consistency, isolation and durability).

Non-repudiation assures that authorized users are not denied access.

Audit is the step of assuring that the access controls are working as intended. This step lets us correct the inevitable mistakes and find attacks.

Security management is the process of setting up the controls.

Who do you trust? How much?

No one?	Security officer?
Administrators?	Auditors?
Application programmers?	End users?

**What software do you trust?**

Operating system?	Security Monitor?
DB2?	Other monitor?
Applications?	Web or PC applications?

The primary job of identification and authentication in DB2 is assigned to the security subsystem. That technique means that access for many resources can be more consistent, whether the resource is a file, a printer, communications or database access. Another way of distinguishing the level of security is the software layer used for the access control. The tightest security would use a single security monitor. Using system controls and subsystems will allow tighter security than having application programs provide the security.

## Primary areas to address for all customers

### z/OS

- system integrity

- security settings

- Security monitor: RACF, ACF2, Top Secret

- DB2 configuration

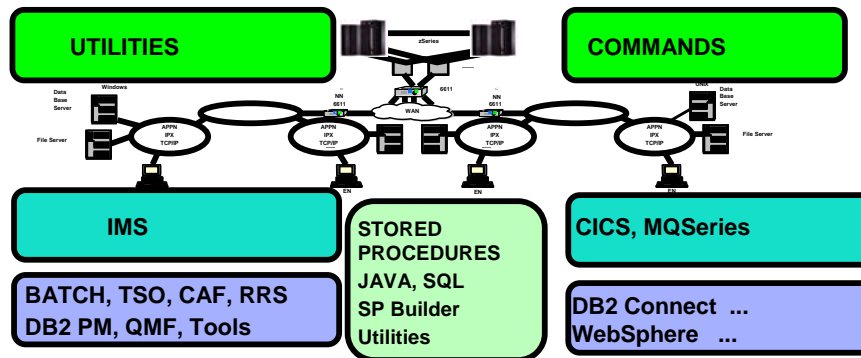
- Encryption

- Auditing

Here are the primary areas for security. I will skip over the sections on the operating system and security monitor. For an example of configuration restriction, see chapter 7, Evaluated Configuration of the Common Criteria in the book, Planning for Multilevel Security and the Common Criteria, GA22-7509-02.

## DB2 Operational Environment

- ➔ Users come from many environments
- ➔ Many possible sources, varieties of userids
- ➔ Many security and audit products, e.g. RACF **Tivoli** software
- ➔ Many options, exits and applications



There are many different environments for DB2, with different connections and security. DB2 uses the security context when possible, so batch jobs, TSO users, IMS and CICS transactions have security that uses a consistent identification and authentication. This is true for stored procedures from these environments as well. The large number of options, exits, environments and asynchronous or parallel work provide challenges for security. Some key applications manage security at the application level.

For some work, such as distributed database serving, DB2 is the initial point on this platform. For this work (DRDA distributed access, remote stored procedures), DB2 establishes the security context and manages the work.

## DB2 configuration

Subsystem parameters

Security exits and interfaces

Encryption options

Protecting data sets

Protecting subsystem connection

Service levels or maintenance

The DB2 configuration has a few parameters with required settings for security, but many other parameters are important to consider. The options, exits and interfaces can make a big difference. Data set protection is provided by the security subsystem. Software currency can also be important for security.

## Subsystem parameters Basic

Use protection DSN6SPRM AUTH YES

Install SYSADM, SYSOPR Groups? Roles? Who?

Other administrators: DBA, OPER, MONITOR2

Security settings for communication

Security settings for routines

Integrating security with other subsystems

identification & authentication

access control

There are a few requirements for basic DB2 security. One is that authorization should always be used. The number of people who can be install SYSADM, SYSADM or SYSCTRL should be small. These names should be groups or roles, and the number of people able to use those groups or roles should be small. My rule of thumb would be 10 people, most of whom do not use this authority for their normal work. Access with SYSADM authority should be audited. Other administrative users should be restricted to the access needed. Where the work is sensitive, auditing is required.

## DB2 UDB for z/OS security exits and interfaces

- ☐ Connection routines and sign-on routines
- ☐ Access control authorization exit routine
  - ☐ RACF access control module
  - ☐ Other vendors
- ☐ Edit routines
- ☐ Validation routines
- ☐ Field procedures
- ☐ Log capture routines
- ☐ Instrumentation Facility Interface or trace
  - ☐ Interpreting trace information
- ☐ Interpreting Recovery Log information

DB2 exit routines can make significant changes in identification, authentication, access control and auditing. Most of the information about these routines is in Appendix B, Writing Exit Routines, of the Administration Guide. Other appendices document tracing, instrumentation interfaces, and recovery log data used for audit.



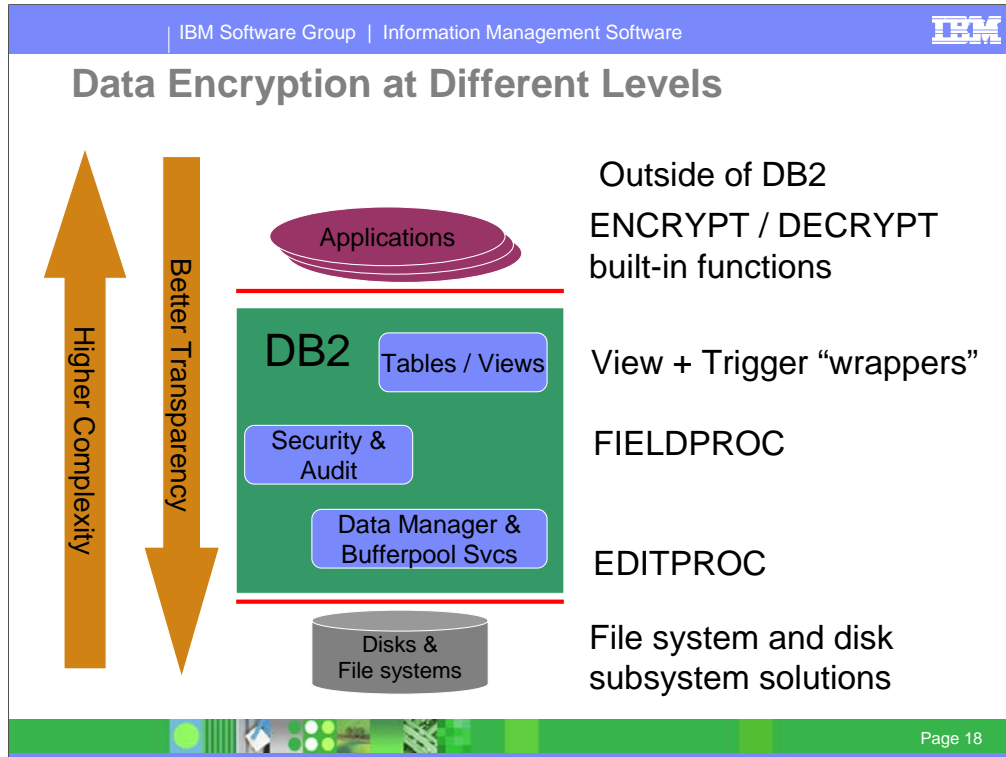
## DB2 and Encrypted Data

What do you want to protect? from whom?  
Techniques, where to encrypt / decrypt

Outside of DB2	General, flexible, no relational range comparisons FOR BIT DATA
DB2 FIELDPROC	No relational range comparisons, FIELDPROC restrictions, FOR BIT DATA
DB2 EDITPROC	indexes are not encrypted, EDITPROC restrictions
User-defined function or stored procedure	General, flexible, invocation needed, no relational range comparisons
SQL functions (V8)	General, flexible, invocation needed, no relational range comparisons
On the wire (DRDA V8, SSL V9)	General, flexible
Tape Backups (z/OS)	General, flexible

There are a number of ways to encrypt data in DB2. The answers to the questions, "What do you want to protect and from whom?" and "How much effort can be used?" are generally needed to determine which technique to use and where to encrypt and decrypt. Encryption does mean some tradeoffs in function, usability and performance. Either the indexes are not encrypted or encrypted data will not give correct results for comparisons other than equals or not equals. All greater than, less than and range predicates are not usable. An EDITPROC is the most used technique, and one is provided to use cryptography hardware with an IBM Tool. The CMOS Cryptographic Coprocessors feature and ICRF hardware were designed to address high volume cryptographic transaction rates and bulk security requirements on z/OS and OS/390. The Integrated Cryptographic Service Facility provides the interface to service routines supported by the hardware, such as key management.

<http://www.ibm.com/servers/eserver/zseries/security/cryptography.html>



This diagram shows the range of places where data encryption can be performed. It is complementary to the prior page, which indicates some of the specific challenges.

If the applications are already written, then there is generally a very high need for transparency. But transparency means that some kinds of protection are not provided.

Some vendors address encryption as well.

## Protecting data sets Basic

Use RACF, ACF2, Top Secret, etc.

Must be done to prevent direct access

DSN1\* usage permitted as required

## Protecting subsystem connection Standard

DSNR access profiles

Any program running on z/OS can access the DB2 data sets, unless they are protected. Define userids for the started tasks and prevent almost every other id from accessing the DB2 data sets. There are some exceptions for administrators who must manage the logs or work with the DSN1\* utilities. Having separate ids for each subsystem is the standard recommendation.

Access control for the subsystem can be used for some separation, for example of test from production.

## Service Levels or Maintenance: Basic

### Later versions

- improved security function
- commands, multilevel, encryption, ...

### Current fixes

- some enhanced function
- service for security exposures (rare)

Service levels for z/OS are generally a bit less critical for security fixes than Unix, Linux or Windows, since there are very few security fixes needed.

Still, later levels and service does provides improvements in security function. Some examples are improved security in DB2 for z/OS V8 with command security, multilevel security and encryption options.

On the Unix, Windows and Linux platforms, be sure to install the needed service.

## Security implementation: Keep it simple

Choice of access control – DB2 or external

How many have administrative access?

Very few SYSADM, SYSCTRL users

Need for DBADM users

Need for SYSOPER, MONITOR2

Use system authority, views, groups, roles ...

Public access only when justified

Be careful with BINDAGENT (weak security)

EXPLAIN access possible without access

One of the keys to success is keeping the security as simple as possible. Having as direct as possible a mapping from the security policy to the implementation will keep mistakes to a minimum, but we must allow for mistakes and for correcting those mistakes.

The number of people who can be install SYSADM, SYSADM or SYSCTRL should be small. These names should all be groups or roles, and the number of people able to use those groups or roles should be small. My rule of thumb would be 10 people, most of whom do not use this authority for their normal work. All access with SYSADM or SYSCTRL authority should be audited. Other administrative users should be restricted to the access needed and also be groups or roles. Where the work is sensitive, auditing is required. SYSOPER can be controlled. MONITOR2 should only be provided to those who can view all work on the DBMS. Public access should be avoided without careful justification and understanding of the security policy.

The BINDAGENT privilege is relatively weak security. Grant BINDAGENT from an id with only the needed authority, not SYSADM in general. There is a fairly new example of how to provide EXPLAIN access when the individuals do not have direct access to the data. An example showing how to have EXPLAIN authorization without direct access to the tables is provided with the V8 Visual Explain and APARs PQ90022 and PQ93821. For this example, you may want to have the binder not have SYSADM, and will not want to grant access to public.

<http://www.ibm.com/software/data/db2/zos/osc/ve/index.html>

## When to look at RACF access control

- **Policy and people implications**

- Roles will change
- Authorities will change
  - ▶ Use RACF facilities more, e.g. groups & patterns
  - ▶ Not a completely compatible change
- Need both DB2 & RACF knowledge for implementation and for administration
- Mix of RACF and DB2 Authorization

- **Security group should define authorization**

- **Centralized Security Control Point**

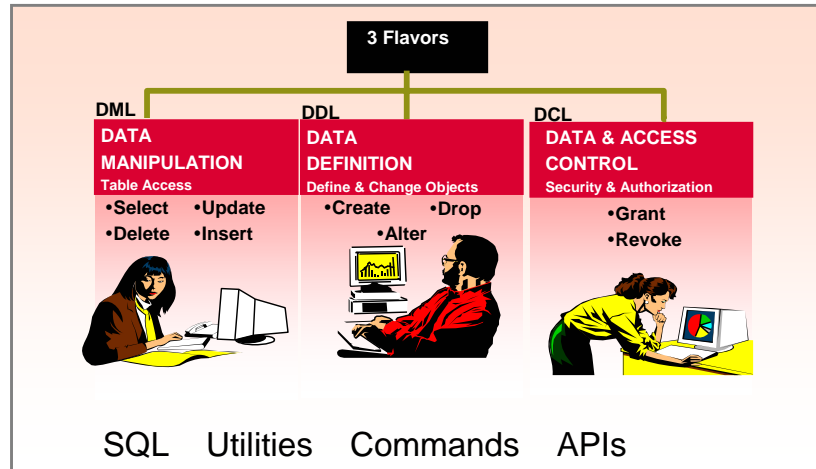
- **Use patterns, not individual access authority**

The choice of using RACF for access control is not for everyone. There are significant policy and people implications. If you want the database administrators to manage security, then integration with DB2 is very important. If you want security administrators to manage security, then integration with the security server is more important. As you make this change, note that roles will change and authorities will change. This is not a compatible change.

You must plan to use RACF facilities more, like groups and patterns. The implementation team needs both DB2 and RACF knowledge for implementation.

If you want a security group to define authorization and a centralized security control point, then this is a match for your needs. As you implement, plan to use patterns instead of individual item access authorities.

## Structured Query Language (SQL)



SQL or the Structured Query Language is generally separated into the ability to manipulate data: get information via **SELECT** or modify via **INSERT**, **UPDATE** and **DELETE**

define data: **CREATE** new objects, **ALTER** them or **DROP** them

data access and control: **GRANT** and **REVOKE** provide the built-in security.

There are many other interfaces into DB2: utilities, commands, and other Application Programming Interfaces (API). Security and authorization are included in **GRANT** and **REVOKE** for all of the interfaces.

## Views

### SQL - Data Definition Language

```
CREATE VIEW SW_CUSTOMER AS  
SELECT CUST_NBR, CUST_NAME,  
CUST_CREDIT  
FROM CUSTOMER  
WHERE CUST_REGION='SW'
```

- Only customers in SW
- Only customer number, name & credit

■ **Views can:**

- Protect data: rows and/or columns
- Simplify access to data
- Join or union to add or remove information

Views can be used to hide data. They can provide only certain fields, as noted. Views are often used to simplify access to data by being able to define the view once and use it many times.

Table privileges DELETE, INSERT, SELECT, UPDATE, or ALL can be granted individually on a view. By creating a view and granting privileges on it, you can give someone access only to a specific combination of data. This capability is sometimes called field-level access control or field-level sensitivity.



## Using a VIEW Basic to Standard

Retrieve from CUSTOMER table  
using SW\_CUSTOMER view

```
SELECT CUST_NBR, CUST_NAME,  
       CUST_REGION  
FROM SW_CUSTOMER  
WHERE CUST_CREDIT = 'AAA'
```

or without the SW\_CUSTOMER view

```
SELECT CUST_NBR, CUST_NAME,  
       CUST_REGION  
FROM CUSTOMER  
WHERE CUST_REGION = 'SW'  
AND CUST_CREDIT = 'AAA'
```

Join to remove information

This example shows the ability to simplify. Using the view, only the additional qualification is needed. Often a view can be used to handle more complex logic, such as a multiple table join or UNION (in Version 7). The person who uses the view does not need to be concerned with the join, UNION or authorization concerns.

## Session Variables: Standard to Best

- Variables set by DB2, connection or signon exit
- Built in function to retrieve value for a variable
  - Use function in views, triggers, stored procedures & constraints to enforce security policy
- Can have more general, flexible access checks
  - Multiple columns, AND/OR logic, ...
- Complements other security mechanisms

```
CREATE VIEW V1 AS SELECT * FROM T1 WHERE  
COL5 = GETVARIABLE('SYSIBM.SECLABEL');
```

Session Variables provide another way to provide information to applications. Some variables will be set by DB2. Others can be set in the connection and signon exits to set these session variables

A new built-in function **GETVARIABLE** is added to retrieve the values of a session variable. This function can be used in views, triggers, stored procedures and constraints to help enforce a security policy. If your primary security need is more general, flexible controls, this information complements other security mechanisms.

For example, you can have a view which provides data that is at the user's current security label.

## Session Variables ...

### Set by DB2 SYSIBM.varname

- PLAN\_NAME
- PACKAGE\_NAME
- PACKAGE\_SCHEMA
- PACKAGE\_VERSION
- SECLABEL
- SYSTEM\_NAME
- VERSION
- DATA\_SHARING\_GROUP\_NAME
- SYSTEM\_ASCII\_CCSD
- EBCDIC
- UNICODE

### Set by connection & signon exits

- Up to 10 variables SESSION.varname

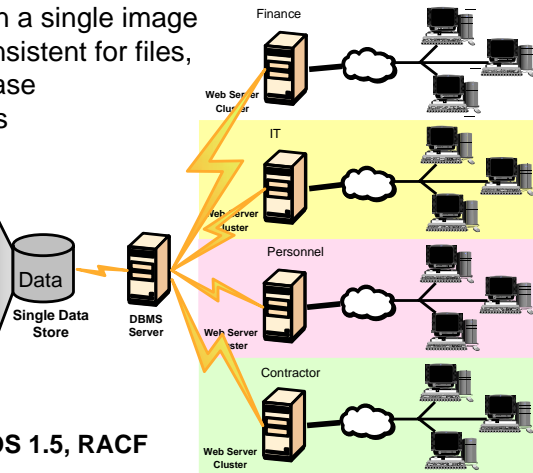
The session variables set by DB2 are qualified by SYSIBM. You can get the plan name, package name (schema, name and version), the user's seclabel, DB2 subsystem name, data sharing group name, version and CCSID information. This information is useful for security controls, but programmers have other needs for this information as well.

Customers can add up to ten variables, with the qualifier SESSION, by setting the name and value in the connection and signon exits. Both the name and the value allow up to 128 characters. Session variables can be accessed, but not changed, in applications.

## Multilevel Security and DB2 UDB for z/OS V8 Best

- Labeled security allows sharing of resources with mixed levels of security in a single image
- Integrated access control, consistent for files, communications, print, database
- Control SQL and utility access

SECURITY LABEL	Col 1	Col 2	Col 3
Personnel	234	USA	50%
Finance	198	France	23%
Personnel	2	UK	9%
Finance	234	USA	11%
Personnel	22	Germany	9%
IT	87	USA	14%
Contractor	23	UK	20%
Personnel	34	Germany	43%
Finance	981	USA	12%
IT	223	USA	10%
Contractor	45	Canada	29%



Multilevel Security on zSeries, z/OS 1.5, RACF

Architecture

Page 28

z/OS 1.5 and RACF or Security Server 1.5 (improved in 1.6) add another type of security, called multilevel security, labeled security or mandatory access control (MAC). The only option in the past with a high degree of separation has been physical separation. In the database world that might mean another machine or LPAR or perhaps another subsystem, another database or another table. With multilevel security, we still have a high degree of security even with data in the same table.

Access control is consistent across many types of resources using RACF, so that multilevel controls apply for data sets, for communications, for print and for database access – both objects and now with row level granularity. The DB2 controls are for both SQL access and for utility access.

For an more on multilevel security, see **Planning for Multilevel Security and Common Criteria (GA22-7509)**

<http://publibz.boulder.ibm.com/epubs/pdf/e0z2e122.pdf>

**Multilevel Security & DB2 Row-Level Security Revealed, SG24-6480**

<http://www.redbooks.ibm.com/redpieces/pdfs/sg246480.pdf>

<http://www.ibm.com/systems/z/security/mls.html>

## Network security:

Use Kerberos, Passtickets, Best  
Encryption of passwords      Basic

Better to encrypt session      Standard or High

Do NOT use already verified      Basic  
Do not trust a client      Basic

Network security is a very important area. The Kerberos and Passticket techniques provide the best security for identification and authentication. If the data is sensitive and uses a network, then protecting or encrypting the network communication is required.

In any case, if you use already verified or trust a client, then the server can be compromised whenever a client is compromised. These are generally unacceptable choices with today's understanding of security.

See the DB2 Administration Guide for much more on this topic.

## Application security Basic → System Best

Access control in application Basic

Use strong system security, views Standard

Static SQL Best

Static authorization rules Standard

Dynamicrules(BIND)

Dynamic SQL – host variables, input checking

Avoid CONNECT with password Standard

Applications do not have some of the protection mechanisms or the level of assurance provided by system security, so use the stronger system techniques whenever possible. Static SQL prevents a number of problems, including SQL injection, while improving performance. Static SQL authorization techniques can be used to avoid granting wide access to tables. If dynamic SQL is used, then use of parameter markers and host variables for input can also avoid SQL injection. Checking the input must be performed. Use of CONNECT with a password provides a shared technique and userid that will make management more difficult. Use system identification and authentication. Changing the password is needed more if you have passwords in programs. There are several vendors for application security function.

IBM Software Group | Information Management Software

IBM

DB2 Audit Data

Basic and up


**DB2 catalog data**

- Tables, Table Spaces, Databases, Views, ...
- Authorization data from GRANT, REVOKE

**Audit Trace sent to SMF, GTF, programs**

- Selective tracing with 9 classes of information
- Access denials
- Authorization changes
- Audit changes, multilevel security
- Update of audited tables
- Access to read audit tables

**DB2 Recovery Log, Image Copies, Data Replication, ...**



Page 31

There are many kinds of audit information available in DB2. The DB2 catalog stores the definitions of all the objects and the authorization. Users who are allowed to access these tables can use the power of SQL to audit and manage security. RACF has an unload facility that allows its security definitions to be loaded into DB2, so that broader auditing can be performed.

One of the primary audit sources is an audit trace that can provide very selective information. Other trace information can also be used in auditing.

The DB2 recovery log and utilities are also helpful in finding out how and when some data was modified.

Please read the Audit section of the Security and Auditing chapter of the DB2 Administration Guide.

## DB2 Instrumentation Records IFCIDs for Audit

### Accounting Audit

0003: successful access  
0140: Audit Authorization Failures  
0141: Audit DDL Grant/Revoke  
0142: Audit DDL Create/Alter/Drop  
0143: Audit First Write  
0144: Audit First Read  
0145: Audit DML Statement  
0314: Authorization Exit Parameters

### Performance

0004: Trace Start  
0005: Trace Stop  
0023: Utility Start  
0024: Utility Change  
0025: Utility End  
0106: System Parameters  
0350: SQL Statement

Here are some suggested audit traces and other traces. The instrumentation has information about the performance, resources, and processes, as well as security and specific sensitive data audits. If more detailed information is required, you can also examine the individual SQL statements (IFCID 0350), record accesses and locking. The data allows full accounting by user/transaction, with detailed data written every plan deallocation or change of user and fully detailed tracing down to individual call / IO / component level.



## Security in DB2 for z/OS Vnext

### Some key implementations

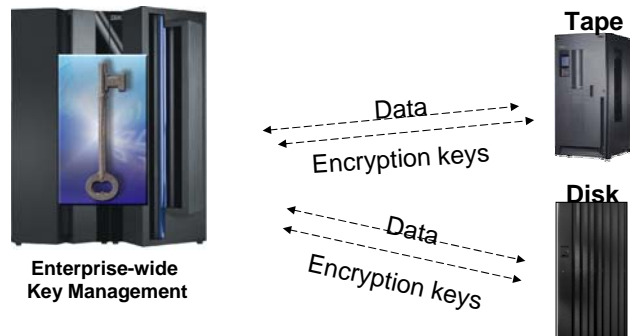
- ☐ Roles
- ☐ Network Trusted Contexts
- ☐ Instead of Triggers
- ☐ Improved auditing
- ☐ Secure Socket Layer
- ☐ Data Encryption



While DB2 for z/OS V8 provides many enhancements for security, there are still many more needs and much more work to do. Roles are used to provide a more flexible technique than groups or users in assigning and controlling authorization, while improving consistency with the industry. A network trusted context provides a technique to work with other environments more easily, improving flexibility. The instead of trigger is an SQL technique that allows a trigger to be used in place of a view, consistent with DB2 for LUW. Improved audit selectivity is needed for being able to see that security is functioning. Secure Socket Layer or SSL implementation provides encryption of data on the wire. Some additional techniques for data encryption will help protect data at rest and in backups.

## Future Directions – Extending Encryption to IBM TotalStorage

- Statement of Direction: To address customers' growing concern with data security, IBM is announcing a statement of direction for the development, enhancement and support of encryption capabilities within storage environments such that the capability does not require the use of host server resources.
- This includes the intent to offer, among other things, capabilities for products within the IBM TotalStorage® portfolio to support outboard encryption and to leverage the centralized key management functions planned for z/OS ICSF.



Statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only

Statement of Direction: To address customers' growing concern with data security, IBM is announcing a statement of direction for the development, enhancement and support of encryption capabilities within storage environments such that the capability does not require the use of host server resources.

This includes the intent to offer, among other things, capabilities for products within the IBM TotalStorage® portfolio to support outboard encryption and to leverage the centralized key management functions planned for z/OS ICSF

This is beyond currently announced products, including DB2 V9.

## Database ROLES

- ROLE is a “virtual authid”
  - Assigned via TRUSTED CONTEXT
  - Provides additional privileges only when in a trusted environment using existing primary AUTHID.
  - Can optionally be the OWNER of DB2 objects

```
CREATE ROLE PROD_DBA;  
GRANT DBADM ... TO PROD_DBA;  
  
CREATE TRUSTED CONTEXT DBA1 ...  
  DEFAULT ROLE PROD_DBA OWNER(ROLE);
```

**Database role:** A database role is a virtual authorization ID that is assigned to the user via the context mentioned next. DB2 privileges are assigned to the defined role.

The role exists as an object independent of its creator, so creation of the role does not produce a dependency on its creator.

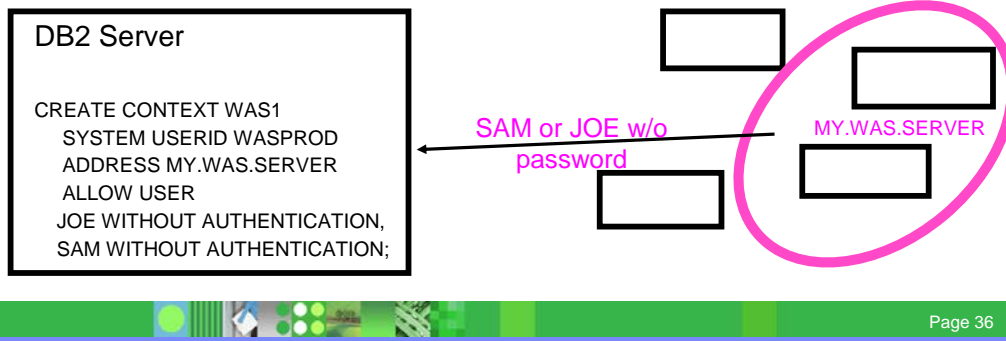
This capability can allow a DBA to have privileges to create objects and manage them for a time, even though ownership is to be another id.

The role can be assigned and removed from individuals via the trusted authorization context as needed. This allows a DBA to perform object maintenance during a change control window on a Saturday night, for example. But when Monday arrives, they do not have the authority to do this same work.

Auditing trails of the work completed during the maintenance window are available for verification by a security administrator or auditor.

## Trusted Security Context

- Identifies “trusted” DDF, RRS Attach, or DSN application servers
- Allows selected DB2 authids on connections without passwords
  - reduces complexity of password management
  - reduces need for an all-inclusive “system authid” in app servers
  - more visibility/auditability of which user is current running
  - enables mixed security capabilities from a single app server



**Trusted security context:** Today, you have the option to set a system parameter which indicates to DB2 that all connections are to be trusted. It is unlikely that all connection types, such as DRDA, RRS, TSO, and batch, from all sources will fit into this category. It is likely that only a subset of connection requests for any type and source may be trusted or that you want to restrict trusted connections to a specific server. More granular flexibility will allow for the definition of *trusted connection objects*.

Once defined, connections from specific users via defined attachments and source servers will allow trusted connections to DB2. The users defined in this context can also be defined to obtain a *database role*.

## Improved audit: DB2 Trace Filtering

- New filtering capabilities for –START TRACE that INCLUDE or EXCLUDE based on these keywords:
  - USERID -- client userid
  - WRKSTN -- client workstation name
  - APPNAME -- client application name
  - PKGLOC -- package LOCATION name
  - PKGCOL -- package COLLECTION name
  - PKGPROG -- PACKAGE name
  - CONNID -- connection ID
  - CORRID -- correlation ID
  - ROLE -- end user's database ROLE

Improved trace filtering makes the jobs of auditing and of performance management easier. Many more options can be used to minimize the amount of data collected, so the overhead is reduced and the extraneous data does not need to be processed.

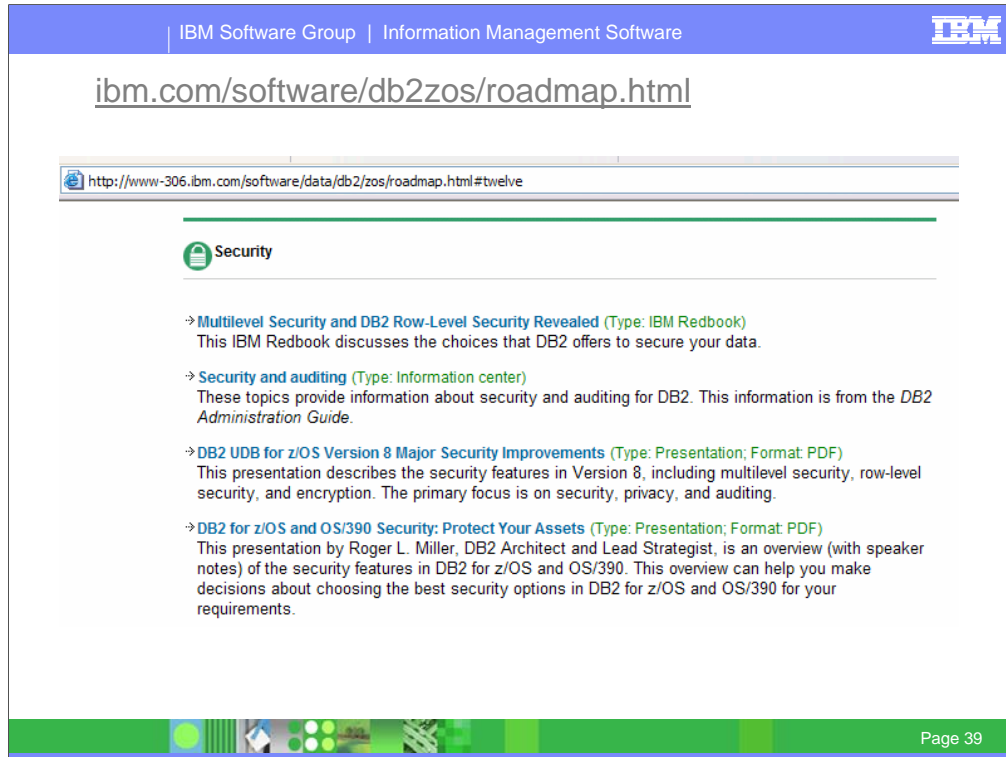
## DB2 Security Provides

### Very significant increased

- ✓ **Security**
  - Mandatory security
  - Row level granularity
- ✓ **Flexibility** 
- ✓ **Integration**
- ✓ **Ease of use for safe security**
- ✓ **Assurance**



Everyone seems to be more aware of security today. Improving integration and making security more robust and easier to manage are very important. Customers asked for a wide range of enhancements for security. New options for tighter security, more granularity, and more information for additional flexibility in applications and SQL are needed. Improved integration with the hardware and software platform provide a better end-to-end solution. The design point for DB2 security is being easy to use for safe security, with solid system security and some security that is more flexible, that we'll call application security. Finally, it will be helpful to see what assurance can be provided, such as certification. DB2 V8 is in evaluation for Common Criteria certification.



The primary information on DB2 security can be found on the DB2 roadmap, after clicking on Security.

<http://www.ibm.com/software/data/db2/zos/roadmap.html>

## References

### ❑ DB2 UDB for z/OS publications:

- Administration Guide, SC18-7413
- Command Reference, SC18-7416
- Data Sharing: Planning and Administration, SC18-7417
- Installation Guide, GC18-7418-00
- RACF Access Control Module Guide V8, SA22-7938
- SQL Reference, SC18-7426
- Utility Guide & Reference, SC18-7427
- DB2 Version 8: Everything you wanted ..., SG24-6079
- Multilevel security and DB2, SG24-6480

### ❑ DB2 information web site:

<http://www.ibm.com/software/data/db2/zos/v8books.html>

Here are some additional pointers for information about DB2 & RACF. One new RACF book is shipped as a pdf file with DB2 and on the DB2 books web page, RACF Access Control Module Guide and Reference Version 8.

Planning for Multilevel Security is on the web under z/OS library, System level books, or

[ibm.com/servers/eserver/zseries/zos/bkserv/r5pdf/zsys.html](http://ibm.com/servers/eserver/zseries/zos/bkserv/r5pdf/zsys.html)

DB2 V8 books including the RACF Access Control Module Guide are located at


<http://www.ibm.com/software/data/db2/os390/v8books.html>



## Disclaimer and Trademarks

Information contained in this material has not been submitted to any formal IBM review and is distributed on "as is" basis without any warranty either expressed or implied. Measurements data have been obtained in laboratory environment. Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion. You should not rely on such information to make business plans. The use of this information is a customer responsibility.

*IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.*

*TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, AS/400, DATABASE 2, DB2, e-business logo, Enterprise Storage Server, ESCON, FICON, OS/390, OS/400, ES/9000, MVS/ESA, Netfinity, RISC, RISC SYSTEM/6000, iSeries, pSeries, xSeries, SYSTEM/390, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS, zSeries, *

*The FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, WINDOWS NT, ODBC, WINDOWS 95*

***For additional information see [ibm.com/legal/copytrade.phtml](http://ibm.com/legal/copytrade.phtml)***

1

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in the operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.