



## Protect Your Information Using New DB2 9 Security Features

Jim Pickel, IBM  
DB2 Development  
[pickel@us.ibm.com](mailto:pickel@us.ibm.com)

Session: RTB11  
Wednesday  
10:30am-11:45am

# Session Agenda

## ❑ Security Challenges

- Many applications contain security controls
- Protecting your information from internal users

## ❑ New Security Features

- Users no longer required to own objects
- Eliminate need to use common IDs
- End-to-end encryption and auditing

## ❑ Best Practices

# DB2 Security Challenges

- Required to control administrative tasks
  - Little control of privileged IDs
  - Little or no individual accountability
  - Difficult to manage or audit database changes
  - Little control of implicit privileges



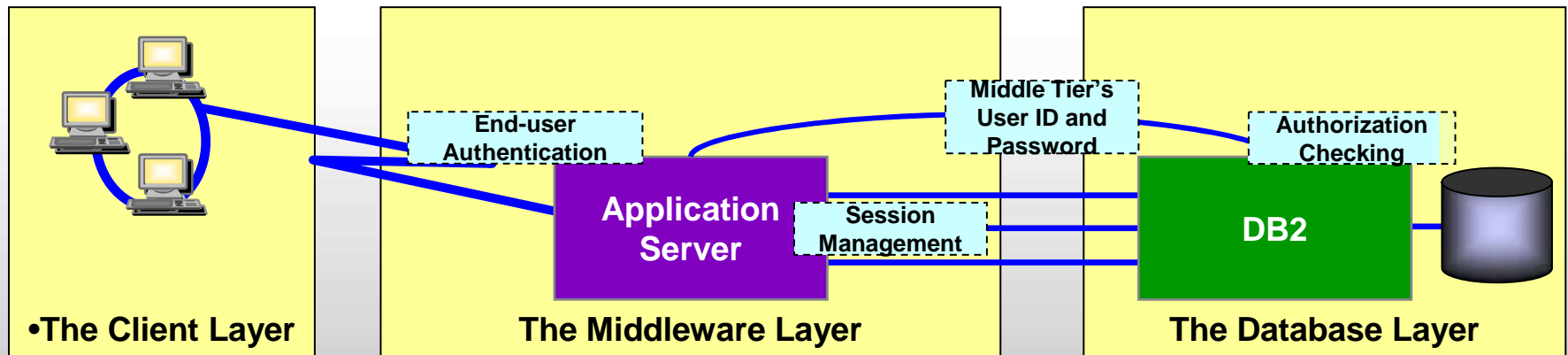
# DB2 Security Challenges

- Need to control application servers
  - Interactions using common IDs
  - Diminished user accountability
  - Over granting of privileges
  - Managing user credentials



WebSphere

# DB2 Security Challenges



- In a typical application server model, the middle layer:
  - authenticates users running client applications
  - manages all interactions with DB2
- The middle layer use a common user ID and password to authenticate connections with DB2
- The common user ID is then used for authorization on behalf of all end-users

# Introducing new DB2 security objects

- DB2 **TRUSTED CONTEXT**
  - A new object used to control users and applications access to DB2
- DB2 **ROLE**
  - A new object that can be granted privileges or own objects



# Associating an application with a trusted context

- Application attributes are verified before associating it with a trusted context such as the system user id and where the request originated
- Allows a unique set of privileges to be associated with an application preventing the misuse of privileges when not accessing through the trusted context
- Controls what end users can be associated with an application eliminating the need to manage RACF user credential from trusted servers

# Using the new ROLE object

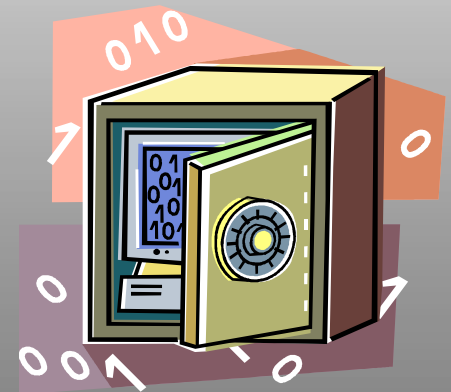
- A role is a object that can be granted any authority or privilege
- A role is only associated with a DB2 process when the application is associated with a trusted context
- A role can be the owner of a database object such as a table
- A role is not a group!





# Other Security Enhancements

- Support User ID propagation
  - Propagates non-RACF users to DB2
  - Non-RACF ID in DB2 and RACF audit logs
  - Minimal CPU impact
  - Requires SAF Enterprise Identity Mapping
- Support end-to-end strong encryption
- Auditing Filtering Improvements



*Satisfies your compliance needs*

# How to Create a Trusted Context and a Role



SQL CREATE/ALTER/DROP TRUSTED CONEXT

SQL CREATE/DROP ROLE

# CREATE TRUSTED CONTEXT

- Provide a system ID and connection attributes necessary to associate a trusted context to a connection
  - ✓ **IP Address or host name of remote application**
  - ✓ **JOBNAME of local application**
  - ✓ **Encryption requirements**
  - ✓ **Enabled or disabled by administrator**
- Provide optional list of users that can be associated with the trusted connection
- Provide authentication requirements for list of users
- Provide optional DB2 ROLE to control application privileges
- Provide optional RACF SERVAUTH profile to control access by network zones
- Provide optional RACF SECURITY LABEL can be associated with the connection



# Trusted Context Example

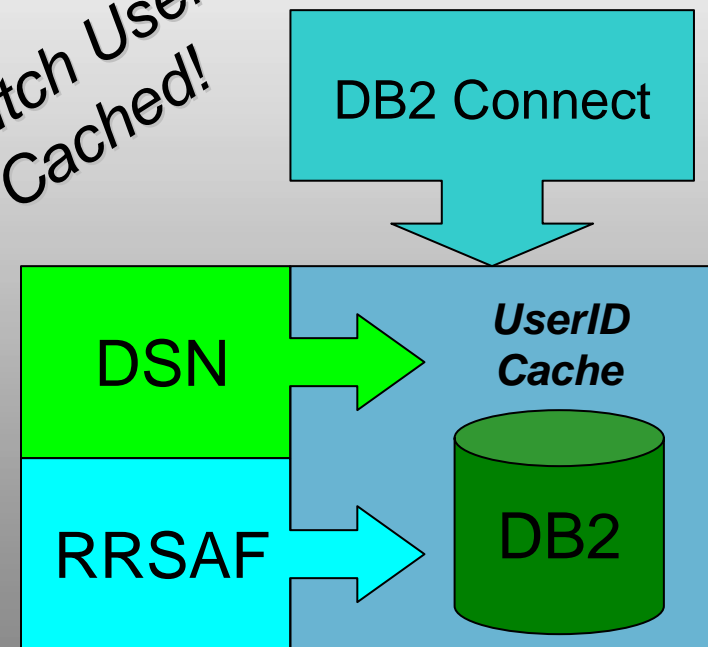
```
CREATE TRUSTED CONTEXT CTX1  
  BASED UPON CONNECTION  
  USING SYSTEM AUTHID WASADMIN  
  WITH USE FOR SAM, JOE, PETE, MARY  
  WITHOUT AUTHENTICATION  
  ATTRIBUTES (ADDRESS '9.67.40.219')  
  ENCRYPTION HIGH  
  SECURITY LABEL SAFEZONE  
  ENABLE;
```

# Establishing a Trusted Connection

- An application can be associated with a trusted context using:
  - ▶ DDF
  - ▶ RRS Attach
  - ▶ DSN ASUSER
  - ▶ BATCH

- Once established, you can securely switch the user associated with connection without requiring credentials

Switch Users  
Cached!



# Client Exploitation

## ❖ New CLI and JDBC Client Driver APIs

- JDBC example:

```
Cookie=getDB2TrustedPooledConnection(sysauthid,  
    syspwd, ...);  
getDB2Connection(Cookie, newUser, newPassword,  
    ...);
```

## ❖ Websphere Application Server

- Database property:

```
propagateClientIdentityUsingTrustedContext
```

# Special Trusted Context Privileges

- Once an application is associated with a trusted context, it can:
  - ▶ Acquire additional privileges through a ROLE
  - ▶ Acquire a RACF security label
  - ▶ Efficiently switch user associated with connection on transaction boundary
  - ▶ Allow objects created to be owned by the ROLE

# CREATE ROLE

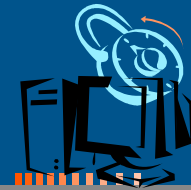
- Creates a DB2 database entity that can have one or more privileges granted to it
- Role associated with a DB2 process when a connection is associated with a trusted context
- Means to acquire context specific privileges
- Can own DB2 objects when trusted context is defined with “Role as Object Owner”





# Role Example

```
CREATE ROLE CTXROLE;  
  
CREATE TRUSTED CONTEXT CTX1  
  BASED UPON CONNECTION  
  USING SYSTEM AUTHID ADMIN1  
  DEFAULT ROLE CTXROLE  
  WITH ROLE AS OBJECT OWNER  
  ATTRIBUTES (ADDRESS '9.67.40.219')  
  ENABLE;  
  
GRANT DBADM TO ROLE CTXROLE;
```



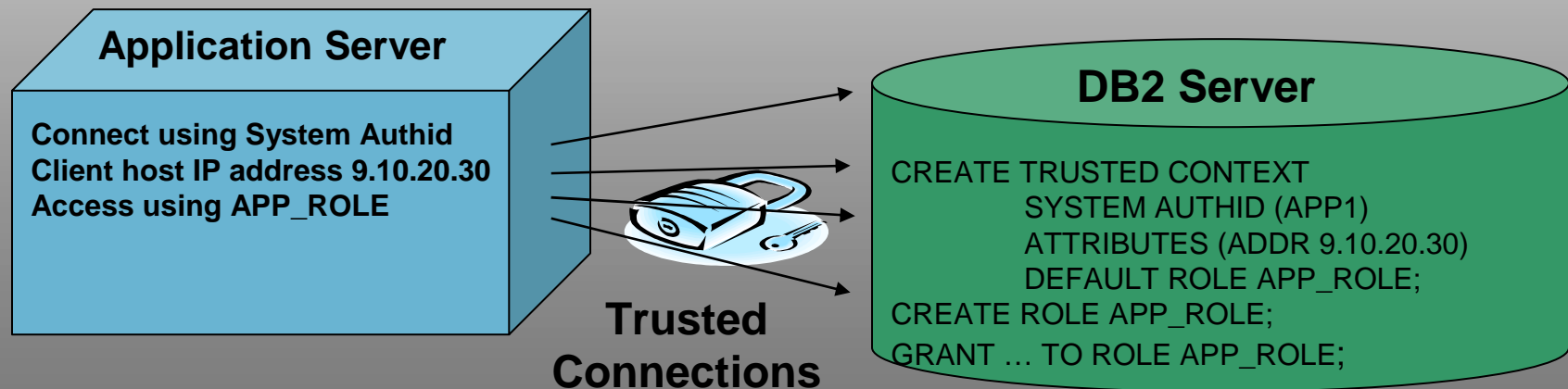
# Best practices using new features

- Secure an existing Application Server
- Secure DBA Activities
- Allow DBA to run as another USER
- Allow remote IDs to be included in z/OS audit logs



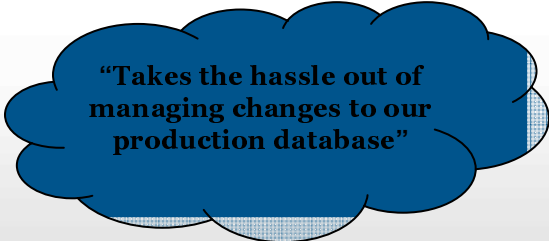
# Secure an existing App Server

- Create TRUSTED CONTEXT, ROLE and associate it with an Application Server
  1. Remove privileges associated with the application server ID
  2. Grant needed privileges to a role used by the application
  3. Change object ownership to ROLE using V9 Catmaint utility
  4. Restrict access to connections from the Application Server IP address
- No changes needed on the Application Server
  - Default Current SCHEMA and Current SQL ID set to ROLE



# Securing DBA Activities

- **Security administrator controls the use of DBADM by**
  - a. Revoking DBA privileges from individual IDs
  - b. Granting special privileges to a DBA role
  - c. Creating trusted context and assign the DBA role to the DBA IDs
- **When a DBA performs a database change, the security administrator then**
  1. Start DB2 audit trace
  2. Enable trusted context to allow access to sensitive objects
  3. DBA can now connect and performs the database change
  4. Disable trusted context to protect sensitive objects
  5. Stop DB2 audit trace
- **An auditor can review the audit trace to ensure compliance**



**“Takes the hassle out of managing changes to our production database”**



# Allow DBA to run as another USER

- For example, a DBADM who created view for other IDs can DROP or ALTER a VIEW owned by the another ID

```
CREATE TRUSTED CONTEXT CTXLOCAL
  BASED UPON CONNECTION
  USING SYSTEM AUTHID PRODDBA1
  ATTRIBUTES (JOBNAME 'DBAJOB*')
  WITH USE FOR PRODOWNR
  ENABLE;
```

```
//DBAJOBA JOB USER='PRODDBA1'
//IKJEFT1B EXEC PGM=IKJEFT1B
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(DB1P) ASUSER(PRODOWNR)
  END
//SYSIN DD *
  ALTER VIEW PRODVIEW REGENERATE;
  COMMIT ;
//
```



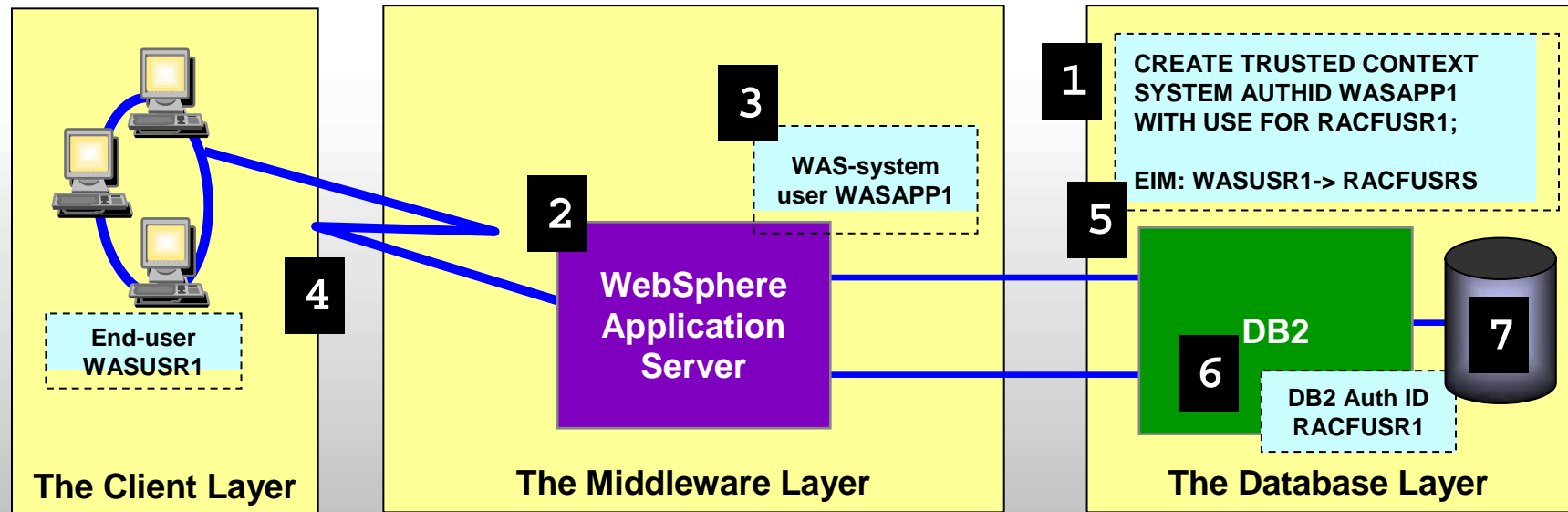
DBA can apply  
emergency changes  
on behalf of the owner

# Identity Propagation using Trusted Context

- Needed when Non-RACF users access DB2
- Non-RACF User IDs included in both DB2 and RACF audit records
- Exploits z/OS Security Server user mapping SAF plug-in service
  - RACF Enterprise Identity Mapping feature (LDAP based)
  - Retrieves RACF Auth ID for remote Non-RACF user ID
  - RACF ID is used primary Auth ID
  - Provides many to one mapping

**End-to-End Auditing!**

# Propagate Identities to DB2 and RACF



1. Configure DB2 to associate a trusted context with WAS
2. Set up EIM WAS user registry (WASUSR1->RACFUSR1)
3. Configure WAS to use a new Trusted Connection API
  - Database property 'propagateClientIdentityUsingTrustedContext' set to 'true'
  - Application parameter 'TargetRealmName' is set to the EIM registry name
4. WAS creates a trusted connection pool using DB2PoolConnection API
5. WAS associates WAS end user with SQL requests using getDB2PoolConnection API
6. DB2 maps the WAS end user ID to obtain RACF auth ID using EIM (results cached)
7. DB2 checks if the DB2 RACF ID is allowed to use the trusted connection
8. WASUSR1 user ID is recorded in both DB2 and RACF audit logs

# Industry Standard End to End Encryption

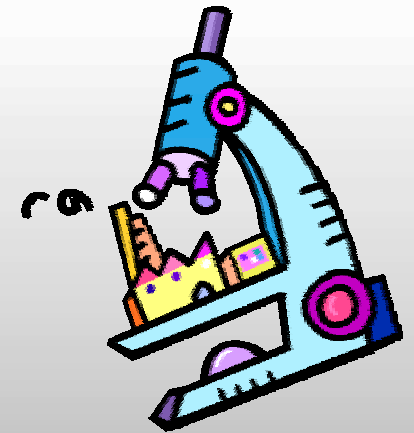
- Default is 256bit AES userid & password encryption
  - Encryption/Decryption runs under DBAT and zIIP enabled
- Support for Secure Socket Layer (SSL) connections
  - Shipped in all IBM data drivers
  - z/OS Communication Server AT-TLS feature
    - Configurable using **Configuration Assistant**
  - DDF can be configured to listen on a new secure port
  - High CPU impact depending on message size
  - IPSEC (VPN) is an alternative and zIIP enabled
- Support for full tape and disk encryption
  - Expanding from tape to disk systems
  - Encrypt data-at-rest with embedded encryption key and password authentication
  - Protects data when disk is removed



# Filter events when using DB2 Audit Trace

- New -START TRACE filtering capabilities that INCLUDE or EXCLUDE audit records based on the following keywords:

- **USERID** – client user ID
- **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** – user's database ROLE



Less evasive audits by allowing auditor to target what records to write



```
-START TRACE ... ROLE(DBA1ROLE,USR1ROLE)  
-START TRACE ... XROLE(DBA2ROLE,USR2ROLE)
```

# Trusted Context and Roles Usage

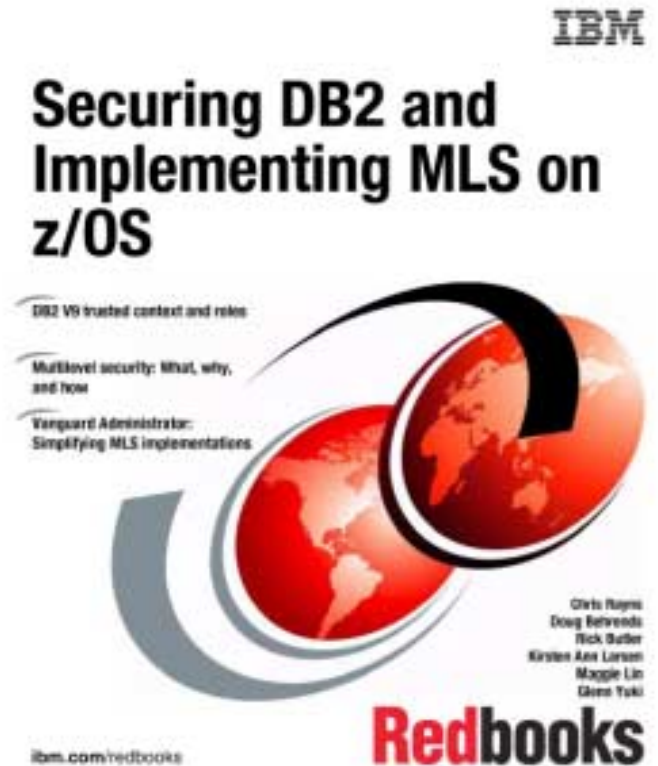
- Better control of application servers
- Better control of administrative authorities
- Removes the need for a user to own objects
- Manage objects owned by other users
- Improved auditing of remote users

# DB2 Security Redbook

SG24-6480

- Updated to include

- Roles
- Trusted Context
- Identity Propagation
- Enabling SSL



# Jim Pickel

[pickel@us.ibm.com](mailto:pickel@us.ibm.com)



## Things to consider when reviewing your DB2 security processes:

1. Incorporate separation of duties in all security processes.
2. Grant only the authority or privileges necessary to do the job (do not over grant)
3. Control the use of implicit privileges by having roles own objects
4. Enable auditing to tables with sensitive data
5. Limit view of data through the views or security labels
6. Control access to DB2 using RACF DSNR and SERVAUTH classes
7. Control applications access through trusted contexts
8. Limit the use of privileged IDs (administrators) through trusted context
9. Prevent the use common IDs by managing users through trusted context
10. Encrypt sensitive data on the disk and on the network
11. Enable identity propagation to allow auditing of end users
12. Perform periodic audits to verify security plan is working
13. Mask test data