Row Level Security with BI Publisher Enterprise

An Oracle White Paper
January 2009

Prepared by
Kanichiro Nishida, Oracle EPM & BI Consulting, Technical Manager
Shankar Duvvuri, Oracle EPM & BI Consulting, Snr Principal Consultant

ORACLE COMMUNICATIONS

ORACLE

NOTE:

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

INTRODUCTION

Having a tight security and clear accountability for reporting data has never been more critical in this more compliance and regulatory requested business world.

Oracle BI Publisher Enterprise provides a framework to increase data security and reduce development and maintenance cost to serve many users with a single report.

Having a tight security and clear accountability for reporting data has never been more critical in this more compliance and regulatory requested business world. A row level security is a technique that allows same report or same query to retrieve different set of data for different users, based on a policy implemented at the data source level. Oracle BI Publisher Enterprise provides a framework to support the row level security that is implemented at the data source and provide a complete enterprise reporting solution to meet today's business compliance and regulatory.

This paper provides an outline of Oracle BI Publisher's row level security support framework and illustrates the implementation procedures required both at the data source and at Oracle BI Publisher Enterprise.

SECURE REPORTING WITH BI PUBLISHER

Many business environments request reports that deliver a different set of data based on who access to the report.

Many business environments request reports that deliver a different set of data based on who access to the report. This is mainly to meet data security requirement. But also this will increase the reporting and data accountability and manageability and reduced the report development and maintenance cost. You can achieve these goals with BI Publisher Enterprise to deliver a secure enterprise reporting.

BI Publisher Security Model Overview

BI Publisher Enterprise provides a full web base reporting access environment where users browse the reports and display the data to analyze their business performance or assist them making better business decision.

It supports a Role-Folder base authentication, which you can assign your users to a role and then assign the role to a folder so that any users who are assigned to the role can access to any reports within the folder.

Business Challenge

While this allows administrators to control who can access to which report, it is not sufficient when you need to control the data inside the report to manage what data to be displayed based on a each user who open the report.

The simple example is a pay slip report that shows employees' salary information for the last salary period. You don't want this report to show all the employees' salary information regardless of who opens the report. You don't want to create the number of employees' reports, each of which will show only respective employee's salary information. This will create a lot of development overheads and maintenance nightmare. You can create a parameter for the report to limit the data. However, this will introduce a possibility that some users might get to see other employee's salary information. For example, a report developer can access to any employee's salary information.

What we want here is to have one single report that can serve different users but depending on which user opens the report it automatically retrieves an appropriate set of data for this user and display it within the generated report.

The best way to achieve this is to implement such data access control policy at the data source level and enable BI Publisher Enterprise to be aware of the policy so that BI Publisher Enterprise automatically retrieves an appropriate data based on the policy and present the data in a generated report.

Solution

In order to address the above mentioned challenge we can take advantage of one of the Oracle database features, Virtual Private Database (VPD), which offers a row level security about the data in the database. BI Publisher Enterprise supports a proxy authentication mechanism, with which it passes session user information down to the database layer and takes advantage of the row level security policy that is implemented at the database to return an appropriate set of data.

What is VPD?

The Virtual Private Database (VPD) enables data access control by user or by customer with the assurance of physical data separation.

The Virtual Private Database (VPD) enables data access control by user or by customer with the assurance of physical data separation. For Internet access, the Virtual Private Database can ensure that online banking customers see only their own accounts. The Web-hosting companies can maintain data of multiple companies in the same Oracle database, while permitting each company to see only its own data.

Within the enterprise, the Virtual Private Database results in lower costs of ownership in deploying applications. Security can be built once, in the data server, rather than in each application that accesses data. Security is stronger, because it is enforced by the database, no matter how a user accesses data. Security is no longer bypassed by a user accessing a reporting tool or new report writer.

How VPD Works

The Virtual Private Database is enabled by associating one or more security policies with tables or views. Direct or indirect access to a table with an attached security policy causes the database to consult a PL/SQL function that implements the policy. The policy function returns an access condition known as a predicate (a WHERE clause), which the database appends to the user's SQL statement, thus dynamically modifying the user's data access.

You can implement VPD by writing a stored procedure to append a SQL predicate to each SQL statement that controls row-level access for that statement. For example, if John Doe (who belongs to Department 10) inputs the SELECT * FROM emp statement, then you can use VPD to add the WHERE DEPT = 10 clause. In this way, you use query modification to restrict data access to certain rows.

Row Level Security with BI Publisher

BI Publisher supports a proxy authentication mechanism that it passes the online user (end user) information down to the data source.

Once the row level security policy with the VPD is implemented for the target tables or views then we can enable BI Publisher Enterprise to be aware of the policy hence it would display only the appropriate data for each user's report access.

BI Publisher supports a proxy authentication mechanism that it passes the online user (end user) information down to the data source so that you can take advantage of such data source level row level security policy implementation.

Not just database but also some other types of data sources support such row level security. For example, Oracle BI Server also supports the row level security so that developers can implement such policies in the business model (or RPD).

In this paper we illustrate the row level security implementation steps with the VPD assuming the data source is Oracle database.

Business Scenario

Let's assume that we have this company which has 3000 employees. End of each month the company sends an email with a link to a report called, Employee Salary. When an employee opens the report he or she should see only their salary related information for the current month period.

Also there is another type of salary related report called, Management Salary report, which only managers can access to and display the salary data for the employees who report to the managers. Therefore, each manager can access to only their employees' salary data.

Row Level Security Implementation Overview

First, we will create two row level security policies with the VPD at a database level. One is for the Employee Salary report and another for the Management Salary report. Next, we will add the policies to a target table that stores the salary

data and verify they work appropriately within the database. Last, we will enable BI Publisher Enterprise to be aware of the row level security policies that are implemented at the database level.

VPD Policy Implementation

The VPD policy can be implemented as a PL/SQL function. All the incoming queries on a table enabled with VPD, are intercepted by the policy function and the incoming query's WHERE clause is altered based on the logic in the policy function. The policy function typically determines the application context and based on that the security requirement is implemented. Once the policy is created you can add it to a target database table to take the policy in effect.

Create VPD Policy

Here is a sample of a VPD policy function that automatically adds WHERE clause to an incoming SELECT query to limit the data based on an employee name.

The policy can be applied to SELECT, INSERT, UPDATE and DELETE statements. Also, one table can have multiple policies associated with it.

```
CREATE OR REPLACE FUNCTION EMP_MAIN_POLICY

(schema IN VARCHAR2, tab IN VARCHAR2) RETURN VARCHAR2 IS

v_empno number;

BEGIN

SELECT EMPNO INTO v_empno

FROM empadmin.managers

WHERE ename = SYS_CONTEXT('userenv','session_user');

IF v_empno = 7839 THEN -- For King President.

   RETURN '1=1';

ELSE

   RETURN 'mgr='|| v_empno;

END IF;

END;

/
```

If you need to exempt certain users from the policy mechanism you can grant EXEMPT ACCESS POLICY to the respective user. In this case the policy function itself will not be executed for the users with the privilege. But this is not recommended for the data governance issue.

Apply VPD Policy

Once you have created required VPD policies now you need to apply them to a target table that you want to apply the policy, in this case it is EMP table. Here is an example of how to add the policy.

```
BEGIN

DBMS_RLS.add_policy

      (object_schema => 'SCOTT',

       object_name => 'EMP',

       policy_name => 'EMP_MAIN_POLICY',

       function_schema => 'EMPADMIN',

       policy_function => 'EMP_MAIN_POLICY',

       statement_types => 'SELECT');

END;

/
```

The above example applies a policy EMP_MAIN_POLICY, which uses the EMP_MAIN_POLICY function that was created above, to the EMP table. When a user submits a SELECT query against the EMP table the query will automatically add WHERE clause based on the definition in the EMP_MAIN_POLICY function.

Once the policy has been applied there is no need to re-apply the policy when the policy function is updated. You can remove the policy with DBMS_RLS.DROP_POLICY function. Please see *Oracle® Database Security Guide* for the detail.

Verify VPD Policy

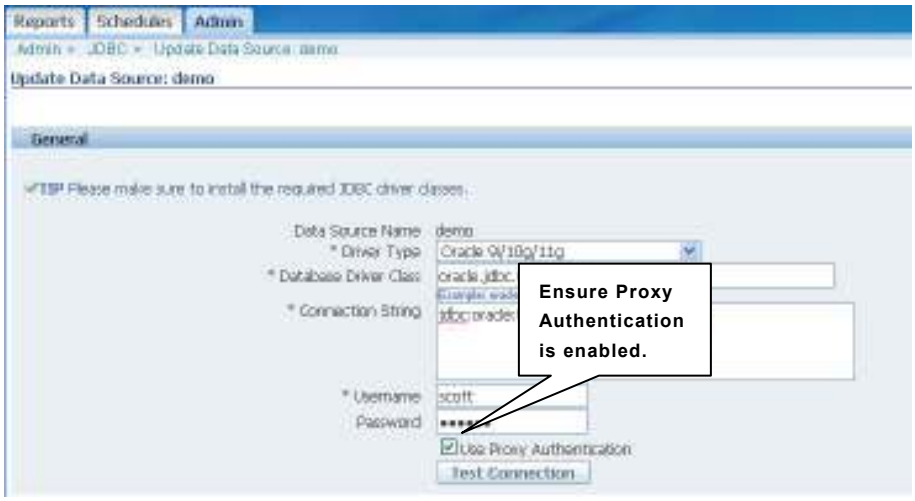
Once all the required policies have been applied to the EMP table it's recommended that you verify the policies work appropriately within the database by using a SQL tool such as Oracle SQL Developer.

Setup at BI Publisher

You need to update your data source setting within the BI Publisher Enterprise administrator page in order to enable BI Publisher Enterprise to be aware of the row level security policies at the database. Also, you need to review your data model that contains SQL queries to adjust with the new row level security setting.

Update Data source

You can login to BI Publisher Enterprise Administrator page and access to the Data Source page where you can enable the Proxy Authentication for each data source. You can simply check the 'Use Proxy Authentication' check box to enable the setting and click the 'Apply' button to take the new setting in effect.



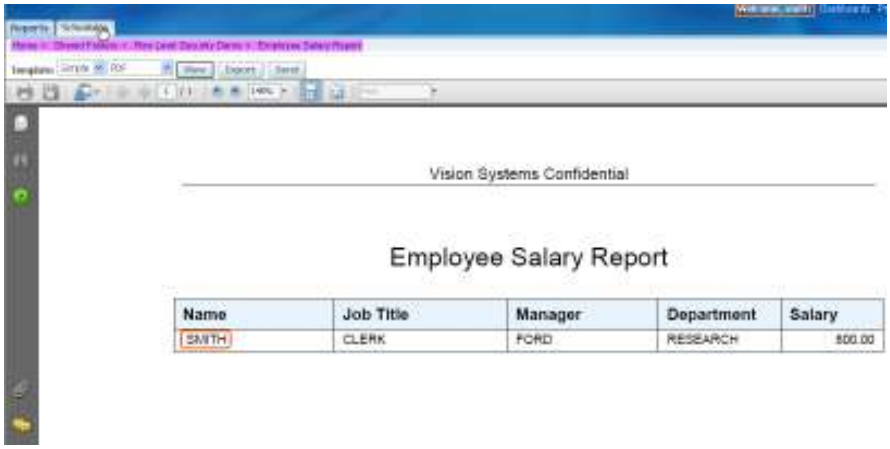
Update Queries for Reports

Since the row level security policies are already in place, any query goes to the EMP table in the database will automatically be updated with an appropriate WHERE condition by the VPD. If your query for the reports contains any WHERE clause that used to limit the data based on the user information then you can now delete it.

Test your BI Publisher report output

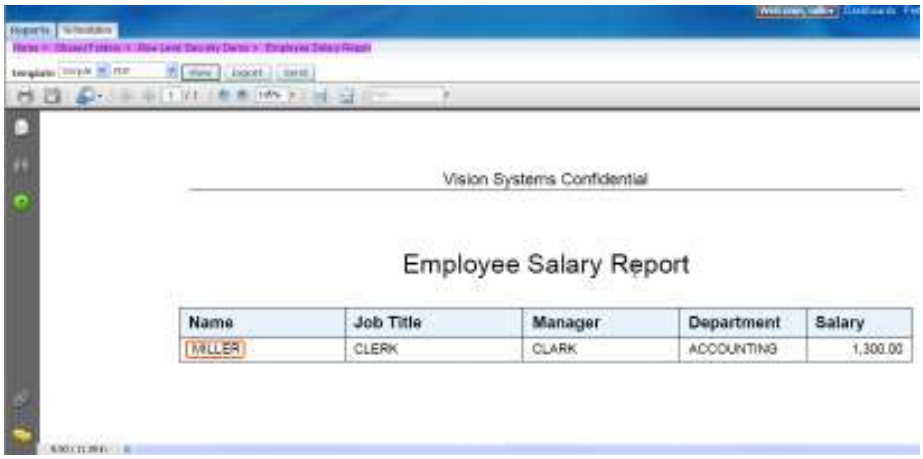
Now you can view the report and see a certain set of data based on whom you've logged in as. For example, if you have logged in as an employee, Smith, when you open Employee Salary report you should see only Smith's salary information and not anybody else's.

Employee Salary Report for Miller



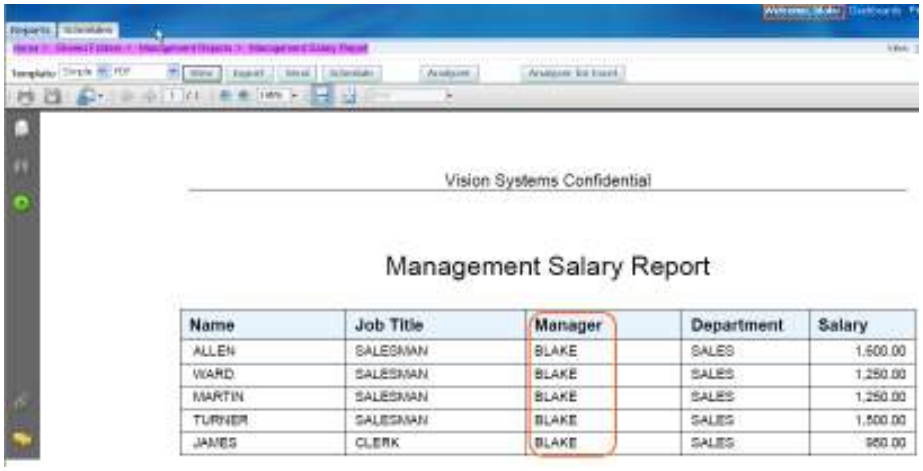
As the same way, if you logged in as Miller you should see only Miller's salary information.

Employee Salary Report for Miller



Also, when you logged in as a manager, Blake, who has five employees reporting to him, then you would see all the five employees' salary information when you view the Management Salary report.

Management Salary Report for Blake



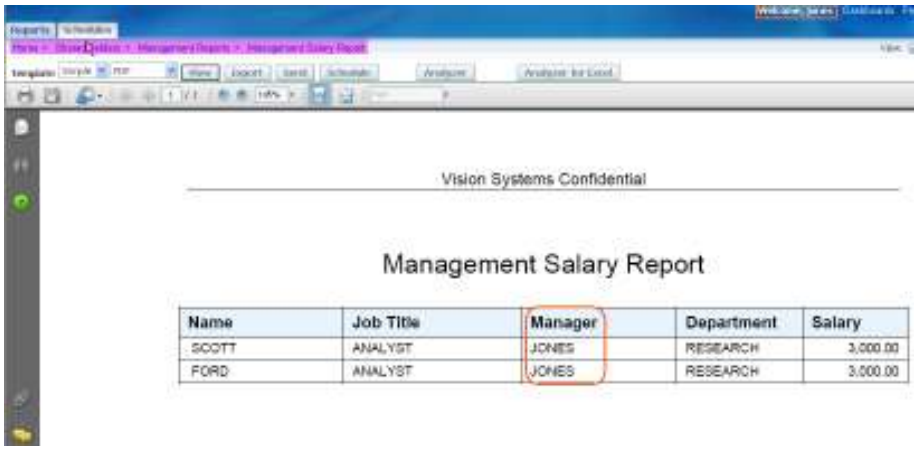
Vision Systems Confidential

Management Salary Report

Name	Job Title	Manager	Department	Salary
ALLEN	SALESMAN	BLAKE	SALES	1,600.00
WARD	SALESMAN	BLAKE	SALES	1,250.00
MARTIN	SALESMAN	BLAKE	SALES	1,250.00
TURNER	SALESMAN	BLAKE	SALES	1,500.00
JAMES	CLERK	BLAKE	SALES	950.00

And when you logged in as another manager, Jones, you would see his only employee information.

Management Salary Report for Jones



Vision Systems Confidential

Management Salary Report

Name	Job Title	Manager	Department	Salary
SCOTT	ANALYST	JONES	RESEARCH	3,000.00
FORD	ANALYST	JONES	RESEARCH	3,000.00

Conclusion

The row level security implementation with Oracle VPD strengthens Oracle BI Publisher Enterprise reporting by providing a tighter data access security and higher data governance in order to deliver a secured and compliant enterprise reporting solution. Not only it shows a different set of data automatically to each user, but also it prevents report developers from viewing any reporting data that they are not supposed see.

The BI Publisher's proxy authentication support mechanism is designed to work with any data source through JDBC or JNDI connection.

The BI Publisher's proxy authentication support mechanism is designed to work with any data source through JDBC or JNDI connection. Once you have setup the row level security policy at your data source including Oracle BI Server you can take advantage of the data source level data management setting with BI Publisher reports. This will allow the report developers to focus on the report design while security administrators can focus on the data access security. And it allows an organization to manage the data access security in a single place instead of having them in many applications and reporting instances.

Our example showed only the row level security with the VPD. But there is another type of data security feature called, Oracle Label Security (OLS), which achieves a similar business goal. With a combination of VPD and OLS you can implement even tighter data access security and higher and more flexible data governance control at the data base level.

Oracle BI Publisher Enterprise is Oracle's strategic enterprise level advanced reporting and publishing tool.

Oracle BI Publisher Enterprise is Oracle's strategic enterprise level advanced reporting and publishing tool. With a combination of Oracle VPD it delivers an enterprise level reporting solution by reducing the cost that was spent for developing and maintaining duplicate security policies and increasing manageability and transparency on the data governance.

Oracle consulting has had many experiences in both row level security implementation and BI Publisher optimized report development. If you're interested in Oracle consulting to discuss more in detail about the implementation and review your reporting data governance and architecture, please contact Kanichiro Nishida (kanichiro.nishida@oracle.com), Consulting Manager, Oracle EPM & BI Advanced Reporting group.

Row Level Security Reporting with BI Publisher Enterprise
January 2009
Author: Kanichiro Nishida

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com