A Survey: Security Perspectives of ORACLE and IBM-DB2 Databases

Lavanya Pamulaparty[*], T. Praveen Kumar[**], P. Vijaya Babu Varma[**]

[*] Associate Professor and Head of CSE, Methodist College of Engineering. & Technology.
[**] Assistant Professor, Dept. of CSE, Methodist College of Engineering. & Technology.
[***] Assistant Professor, Dept. of CSE, Methodist College of Engineering & Technology.

Abstract- As storage of data plays an integral part of databases, security issues becomes major concerns. Relational databases hold a significant portion of data stored in software, therefore today's database purchase decisions revolve around how secure the product is. This paper provides a categorical feature comparison between Oracle9i Database (Oracle) and IBM DB2 Universal Database (DB2), in addition to examining features provided in the SecureWay product line from Tivoli, an IBM subsidiary [7]. It explores the impact of IBM's and Oracle's security models on users seeking to protect their critical information systems and contrasts IBM's strategy of building security outside of the DB2 database against Oracle's strategy of securing information in the database server[6].In addition to the security issues we explore some of the strategical issues arises in database migration.

Index Terms- Database security, Oracle Security, DB2 Security, Tivoli SecureWay, Database Migration [6].

I. INTRODUCTION

IBM and Oracle differ sharply in their fundamental approaches to security. On one hand, Oracle endeavors to build security features and solutions into each of its products, particularly the database server, where data is stored. This approach means that customers get out-of-the-box security when they install and configure Oracle. Security is at the core of the coding practices employed by the development staff that builds the Oracle database, resulting in the delivery of a secure product [7]. Oracle recognizes that they must ship a certified, provably-secure database. Such assurance is afforded by independent security evaluations against established security criteria. Assurance is a large part of Oracle's approach to security, and it differentiates Oracle from other database vendors. On the other hand, IBM addresses security by delivering it outside of the database and relying on the operating system or Tivoli's product line to secure DB2 and other IBM products. The most obvious result is that data stored in DB2 is not inherently protected; one must deploy Tivoli SecureWay products to protect DB2 [6].

Another outcome is that IBM's strategy interjects IBM Global Services into security purchases because service is often required to integrate the DB2 and Tivoli product sets. These outcomes have financial implications as well: customers must spend additional dollars on Tivoli products to secure DB2, and IBM Global Services involvement increases the cost of implementing security in a DB2 environment. Further, IBM lacks independent assurance of the security built into DB2. Whereas Oracle has undergone multiple evaluations of its database, IBM has failed to have independent experts formally evaluate DB2, making it difficult to qualify their assertions about their security implementations. Oracle's business model is to secure products out-of-the-box, and IBM's is to make customers pay to secure the products they purchase. This divergence in approach demonstrates the value of security to these database competitors and the resulting security built-in to their customers' deployments [7].

II. IMPACT ON CUSTOMERS

IBM's Approach towards Customers

IBM's security business is solid. They understand security, participate in standards committees, and, in fact, IBM researchers developed the Data Encryption Standard (DES). The security model they choose to secure the database, however, has flaws that impact their customers. The DB2 security model favored by IBM hurts customers in three ways:

- A less secure database, more vulnerable to users or hackers subverting the security due to the security model that adds security after the fact. It is difficult to add layers of security after a product has been designed, coded and shipped [7].
- Higher up-front costs because of the additional products necessary to secure DB2. Customers must purchase a database that includes little out-of-thebox security, then augment the purchase with other products.
- Higher long-term cost of ownership because customers must pay for the database product, the security product and required services— plus upgrades and support services for multiple products over the years.

Oracle's Approach towards Customers

Oracle has an excellent, long-standing reputation in security, as witnessed by Oracle's dominant market share among the most security-conscious customers in the world. The Oracle security purchase is more straightforward than that of IBM because Oracle integrates security features into each of its products. The Oracle9i Database (both the Standard and Enterprise Editions) provides industry leading security features in the products, rendering it difficult to subvert security. Unlike DB2, Oracle security stands on its own without requiring customers to license products for such advanced features as granular access control and customizable auditing (though Oracle provides security options to further enhance its security

offerings). The feature-for-feature comparison later in this paper substantiates this point. Further, independent security evaluations examine the security of Oracle without extra-cost options. These independent evaluations validate the Oracle database itself, without the help of features supplied in add-on options. Finally, because Oracle includes security functionality, Oracle's customers are not obliged to purchase add-on products for fundamental but essential security features, nor must they pay for upgrades and support for such additional products. [6] The following table summarizes the impact on customers of the two companies' divergent approaches

Table.1 Impacts on Customers

IBM	Oracle
Security outside of database makes DB2 more vulnerable to users subverting security.	Oracle provides industry-leading security features within the database product, rendering it difficult to subvert security.
Customers purchase a database with little out-of-the-box security, then augment the purchase with security products. Required products and services result in higher up-front prices.	Oracle database security stands on its own without requiring customers to license separate security products for essential, evaluated security features.
No independent validation of DB2.	Independent security evaluations validate proper implementation of security in the Oracle RDBMS.
High long-term cost of ownership because customers must pay for the database product, security products and required services— plus upgrades and support services for all those products.	Customers are not obliged to purchase add-on products for key security features, nor pay for upgrades and support for such products.

III. State of Security in Oracle and DB2

A. Feature Comparison

To best understand Oracle versus IBM security, let's look at a feature-for-feature comparison of their complete offerings. Because IBM builds little security into the DB2 database products, the comparison takes into account features in the DB2 family of database servers, the Tivoli SecureWay product line, as well as those supplied by the OS. On the Oracle side, the comparison looks at security features included in the database license, along with features provided by extra-cost database options. [9]

B. User Authentication

The basis for system security is strong user identification and authorization. If you cannot establish, with certainty, who a user is, then it is impossible to hold users accountable for their actions, and difficult to ensure that users only have access to the data they need to do their jobs, but no more. DB2 provides basic authentication and authorization support. Installation requires the administrator's username, password, and group name (and DB2 provides a default for each of these to the user doing the install). Users are defined by user ID in DB2 or the underlying operating system, and IBM supports most of the popular authentication methods. That is, users can be authenticated using DB2 passwords, by relying on the server, the operating system, Kerberos, or Distributed Computing Environment (DCE) credentials. Oracle supports a number of choices for user authentication: Oracle-based (by password, or by industry-standard digital certificates), host-based (by the underlying operating system), or third-party based (network authentication services Kerberos, CyberSafe and DCE, token cards, smart cards and biometric devices).7 Oracle provides built-in password management facilities to enable administrators to enforce minimal password length, ensure password complexity, and disallow passwords that are easily guessed words. Both IBM and Oracle provide adequate basic user identification and authentication support[3].

C. Authorization and Access Control

Privileges

A user's authorizations determine what data he should have access to and what types of operations he can perform on those objects. A user can only perform an operation on a database object (such as a table or view) if that user has been

authorized to perform that operation. A privilege is an authorization to perform a particular operation; without explicitly granted privileges, a user cannot access any information in the database[1]. To ensure data security, a user should only be granted those privileges that he needs to perform his job functions. This is known as the principle of "least privilege." To ensure data security, both DB2 and Oracle use authorizations to enable users to access the appropriate database objects and resources. Both use the same definition of privileges and use standard SQL. For example, to assign Scott the select privilege on the employee table in DB2 or Oracle, the syntax is the same: grant select on employee to user scott Both databases enable a grouping of privileges in roles. [5]

Views for Access Control

Views allow you to further limit the data that a user can access within an object. A view is a subset of one or more tables (or views). You can define, for example, a view that allows a manager to view only the information in the employee table that is relevant to employees in her own department. The view may contain only certain columns from the base table (or tables), such as employee name and salary. Views can also limit the subset of the rows accessible in the base table, such as a view of the employee table which contains records for employees assigned to department 10 [1].

Granular Access Control

A foundation of security is controlling access to data. Who would consider opening production systems, such as order entry, inventory and customer support, to customers and partners without the ability to strictly limit data access? Internet-based systems have a strong requirement for access control at a very fine level of granularity, often to the level of individual customers or users [2].

Virtual Private Database

In 1999, Oracle8i set a new standard in database security with the introduction of Virtual Private Database (VPD), unique to Oracle. The Virtual Private Database enables, within a single database, per-user or per-customer data access with the assurance of physical data separation. VPD is the aggregation of server-enforced, fine-grained access control, together with a secure application context in the Oracle database. By dynamically appending SQL statements with a predicate, VPD limits access to data at the row level and ties the security policy to the table (or view) itself. 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 utilizing an ad hoc query tool or new report writer. [8]

Examples of VPD customers include:

• Several large banks and financial services companies use it to separate customer or employee access to financial data.

• Security-conscious U.S. Federal government organizations use it for even the most rigid implementations.

• A financial services company uses it to apply a set of rules based on user identity and position in the organization.

IBM has no comparable feature set beyond its basic authorization and access control mechanisms (the very features Oracle felt were not enough for today's demanding customer requirements). Neither Tivoli's security applications nor IBM's operating systems provide such functionality. This is one area in which IBM Global Services may get involved to develop custom code. "Custom code developed by IBM allows [the customer] to monitor which users access case documents.

RACF

DB2 takes advantage of Resource Access Control Facility (RACF) for access control in a mainframe environment. Without RACF underlying other DB2 databases, such as in the DB2 product for Unix/NT/Linux, administrators cannot secure all instances of DB2 in the same way. When the software does not natively support a feature or service, and this is a fine example, IBM relies on Global Services consultants to custom build a solution for the customer. RACF on the mainframe augments Oracle's internal database security because Oracle supports RACF for customers running the Oracle database on mainframes [7].

D. Encryption

The Internet poses new challenges in information security, and encryption leads the pack of solutions used to address the traditional list of security threats. It is becoming more important every day to encrypt especially sensitive data in the database as well as packets flowing over any network. [2]

Encryption in the Database

IBM has delivered an introductory database encryption capability in the most recent release, DB2 UDB 7.2, available since June 2001. DB2 has functions that enable an application to encrypt and decrypt data using an RC2 block cipher with a 128-bit key and using an MD2 message digest. It provides column-level encryption, enabling all values in a column to be encrypted with the same key— an encryption password. First delivered in Oracle8i in 1999, Oracle provides an encrypt/decrypt interface to encrypt especially sensitive data in the database server. Oracle has been enhancing the database encryption solution over the years, adding in Triple-DES encryption and MD5 cryptographic checksums in a subsequent Oracle8i release. The first Oracle9i release enhanced the Random Number Generator (RNG) to use a FIPS 140 Level 2-certified RNG, another example of security with assurance. In the current release, Oracle provides DES (56-bit), 2-key and 3-key Triple-DES (112- and 168-bits, respectively) in an encryption toolkit package that enables applications to encrypt data within the database. The IBM solution is password-based; the user supplies a password as the encryption key to encrypt and decrypt data. This is an elegant solution; however it does have certain drawbacks. First, there has been no independent certification of implementation (e.g., FIPS 140). Second is implementation. While there is a minimum password length, DB2 SQL Reference documentation warns, "It is the user's responsibility to perform password management"10 because there's nothing to stop a user from never changing a weak password which may be susceptible to a dictionary attack. [6]

Network Encryption

DB2 database itself does not provide network encryption to secure communications between any client and the database, but IBM does support DES and RC2 in the network. For example, IBM encrypts the network in the z/OS mainframe, has an OS/390 Virtual Private Network, and the Tivoli Management Framework supports SSL and DES. Customers must purchase additional IBM products to encrypt various network layers, but with the appropriate products in place, they can secure the network on which DB2 sits. Oracle offers Oracle Advanced Security to protect all communications with the Oracle Database. Wherever the database is available, Oracle9i Advanced Security is available and ships on the same media as the database software. To encrypt network traffic, it provides Secure Sockets Layer (SSL). [11] the Internet standard, and offers:

• RC4 in 256-bit, 128-bit, 56-bit, and 40-bit key lengths,
• DES in 56-bit and 40-bit key lengths,
• 2-key or 3-key Triple-DES (3DES) with 112-bit and 168-bit keys, respectively, which is especially high-strength encryption.

These cryptographic modules have undergone the laborious certification process to claim Federal Information Processing Standard (FIPS 140-1) Level 2 compliance, providing assurance of the implementation— down to the randomness of key generation. To prevent modification or replay of data during transmission, Oracle uses an MD5 or SHA-1 message digest included in each network packet. The encryption and data integrity capabilities protect Oracle clients and middle tier servers in communications over Net8, Net8/SSL, IIOP/SSL, and also secure Thin Java Database Connectivity (JDBC) clients. In short, Oracle provides a variety of ways to encrypt communications over all protocols with any database communications. Wherever the database runs, the network traffic can be protected with encryption. IBM and Oracle take different approaches to securing network traffic. Oracle's implementation is tied more closely to its database, but both provide ample solutions for the demanding customer requirements stemming from the susceptibility of clear text data flowing over corporate networks, intranets, and the Internet. [15]

E. Auditing

Auditing is a passive, albeit important, security mechanism. A critical aspect of any security policy is maintaining a record of system activity to ensure that users are held accountable for their actions. To address this requirement, both DB2 and Oracle provide extensive audit facilities. [10]

Fine-grained Auditing

Fine-grained auditing allows organizations to define audit policies, which specify the data access conditions that trigger the audit event. Administrators can use a flexible event handler to notify them that the triggering event has occurred. For example, an organization may allow HR clerks to access employee salary information, but audits access when salaries greater than $500K are accessed. The audit policy ("where SALARY > 500000") is applied to the EMPLOYEES table through an audit policy interface (a PL/SQL package). In addition, the event handler sets a triggering audit event to be written to a special audit table for further analysis, or it could activate a pager for the security administrator. DB2 offers no support for such granular and customizable auditing. In general, auditing does not capture the data returned to the user because audit logs would become too large. Fine-grained auditing captures the exact SQL text of the audited statement, and when used in combination with Oracle's Flashback Query feature, you can recreate the exact records returned to a user. This combination defends against the user who tries to subvert the auditing mechanisms by issuing hard-to-detect queries that may hide the intent of the query.

Oracle produces a graphical user interface tool, Oracle Selective Audit, to automate auditing management and analysis. The tool integrates auditing with database logs, LogMiner, and Flashback Query to capture and display all relevant queries. It provides a graphical way to detect suspicious activities, such as a user attempting to login as administrator after hours or accessing more data than he should because a DBA inadvertently assigned him incorrect privileges. With the click of a mouse, auditors can view DDL and DML statements, view the exact SQL text issued, and even play back rows returned to the user at the time of the query— even if the database has changed dramatically since the issuing of the query. No database vendor apart from Oracle offers such a comprehensive auditing picture.

SecureWay Auditing

SecureWay Security Manager and SecureWay PKI are Tivoli products which provide auditing facilities to enhance the auditing features in DB2. SecureWay Security Manager audits user login and access to various resources, and it presents audit reports to the auditor. It enables auditors to log, view, and report security administrative actions.13 SecureWay PKI, in addition to providing PKI services, creates a separate audit trail of administrator activities. These auditing capabilities in the Tivoli SecureWay product line are useful additions to the IBM's DB2 auditing story. Oracle and IBM both provide a host of auditing solutions, though the scope and granularity of auditing features shipped inside Oracle9i Database leads all of its database competitors. Customers with a need to log and inspect database access without taking on high overhead, those with corporate auditing mandates, and those with industry regulations (such as HIPAA in health care) use these advanced auditing capabilities innovated by Oracle.[7]

Table.2 Security Features

Feature or Area	Oracle	IBM-DB2	Tivoli SecureWay
Authorization	Yes	Yes	Yes
Basic Auditing tools	Yes	Yes	Yes
Fine-grained Auditing	Yes	No	No
Data Encryption	Yes	Yes	Not Applicable
Fine-grained Access Control	Yes	No	No
PKI Support	Yes	NO	Yes
Evaluated RDBMS	Yes	No	Yes

RACF Support	Yes(No MainFrames)	Yes	Yes
Network Encryption	Yes	No	Yes

F. Migration Strategies

Migration from Oracle Database to IBM DB2 is not completely seamless and must be planned carefully. Administrators may face issues when migrating from Oracle to DB2 due to locking differences between the two databases. However, these issues can be mitigated to a great extent. One of the key locking behavior differences between Oracle and DB2 is that Oracle does not hold any locks on a row while reading, and DB2 does. This difference can lead to a high probability of increased lock waits and issues such as deadlocks and timeouts in applications migrated from Oracle to DB2. To handle locking issues, mitigation strategies are required at the database, application, and operational levels.

Database-level strategies

Several types of database design changes can help mitigate locking issues:

- ***Row-level locks***. Override the default DB2 page-level lock setting and reset so that the table uses row-level locking to increase concurrency. Row-level locks should be implemented carefully, since there could be increased overhead due to the growing number of locks, and the potential for lock escalation increases if not properly handled.
- ***Index and query tuning***. Read queries, which might require a table scan, would not cause a problem in Oracle but would be an issue in DB2 on z/OS because they would lock the entire table. To mitigate this problem, ensure that all queries are optimized in terms of index and access path so there are no unnecessary table scans, especially for tables that are accessed in online transactions.
- ***Partitioning.*** Concurrency can be increased a great deal, especially for batch runs, by introducing partitioned table spaces in DB2 for z/OS. Data can be segregated into different partitions by identifying a partitioning key and having the data reside on different partitions based on the range of values of the key. When running a batch, multiple threads can be initiated based on the partitioning key value, so that the different threads access different partitions and provide higher concurrency.

Application-level strategies

Some of the key application design changes that can help mitigate locking issues include:

- **Skip locked data.** You may have a situation in which different transactions are going against the same table and you need to access only the rows that are not currently locked in any given table. In these cases, DB2 provides an option to query only the rows that are not locked by using the SKIP LOCKED DATA option in the SELECT, UPDATE, and DELETE clauses. This option applies only when the isolation levels of Cursor Stability (CS) and Read Stability (RS) are in place and also applies only to row-level and page-level locks.
- **Uncommitted read.** In cases where it is acceptable for the response from a read query to have uncommitted data, try using the WITH UR option in read queries in DB2, since this does not hold any shared locks. This option is very helpful for user queries run by application testers or business analysts in the user acceptance testing or production regions. These queries could contend with application queries, which can be avoided by running the user queries using the WITH UR clause.
- **Table access ordering.** Increases in locking contentions can also occur when migrating from Oracle to DB2 for z/OS due to improper ordering of access to tables in parallel transactions. Consistent access ordering can help avoid this problem. For example, if transaction 1 accesses table A first and then table B, subsequent transactions should use the same order when accessing the same tables.

Operational-level strategies

Contention can occur due to different types of workloads going against the same table—for example, batch and online workloads accessing tables at the same time, or different batches accessing the tables at the same time. In these scenarios, one option is to make operational-level changes such as rescheduling the conflicting transactions. It may be possible to run a batch at off-peak hours when the online workload is not running. In case of two batches running parallel, try running one after the other, or put dependencies in place so that one cannot run when the other is running, and vice versa.

IV. Conclusion

At first glance, Oracle and IBM appear to offer similar security solutions, but with closer inspection, it is plain to see that the two companies approach security differently and ship solutions at vastly different levels of maturity. Independent evaluations and feature-for-feature comparisons prove that the Oracle9i Database is more secure than IBM's DB2 Universal Database. Overwhelming evidence supporting this assertion, as established in this paper, proves that Oracle security is far superior to DB2 security. The Oracle database builds-in security and stands on its own; the database itself has achieved nine independent evaluations performed by industry experts. IBM has not completed any evaluations of DB2. While IBM has a good reputation in security in general, they provide no independent gauge of DB2 security implementations. There are several key locking differences between Oracle and DB2 on z/OS that can lead to locking issues with applications migrated from Oracle to DB2. However, this paper demonstrates options that are available at the database, application, and operational strategic levels to greatly mitigate any issues that might arise.

REFERENCES

[1] Rashid, Z.; Basit, A.; Anwar, Z.; "TRDBAC: Temporal reflective database access control" 6th International Conference on Emerging Technologies (ICET), 2010

[2] Zheng-Fei Wang; Ai-Guo Tang; "Implementation of encrypted data for outsourced database" Computational Intelligence and Natural Computing Proceedings (CINC), Volume 2, 2010

[3] Yan Zhao; Yongcheng Luo; Jian Wang; Jiajin Le; "A novel privacy preserving approach for database security" ICTM Volume 1, Pages 408-411, 2009.

[4] Anbalagan, P.; Vouk, M.; "Towards a Unifying approach in Understanding Security Problems", ISSRE, pages 136- 145, 2009.

[5] Michel Abdalla, Emmanuel Bresson, Olivier Chevassut, Bodo Möller and David Pointcheval, "Strong Password-Based Authentication in TLS using the Three-Party Group Diffie-Hellman Protocol", International Journal of Security and Networks, vol. 2, numbers 3/4, pp. 284-296, Inderscience, 2007.

[6] OracleTechinicalDocumentation - http://www.oracle.com

[7] IBM-DB2 Techincal Documentation - http://publib.boulder.ibm.com/infocenter/

[8] Shehab, M.; Bertino, E.; Ghafoor, A.;" Watermarking Relational Databases Using Optimization-Based Techniques" IEEE transactions on Data Engineering, Volume 20, Issuer 1, 2006.

[9] Bertino, E.; Sandhu, R.;" Database security- concepts, approaches, and challenges"IEEE transactions on Secure Computing, Volume 2, Issue 1, pages 2-19, 2005.

[10] Ramasubramanian, P.; Kannan, A.;"An active rule based appraoch to database security in e-commerce systems using temporal constraints", TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region, Volume 3, pages 1148-1152, 2003.

[11] D.Taylor; "SSL for TLS Authentication"IETF draft- ietf-tls-srp-01.txt (work in progress) June 29, 2001

[12] S. Halevi and H. Krawczyk;"Public-key cryptography and password protocols" ACM Transactions on Information and Systems Security (TISSEC), Vol. 2, August 1999.

AUTHORS

First Author – Lavanya Pamulaparty, Associate Professor and Head of CSE, Methodist College of Engineering. & Technology.
Second Author – T. Praveen Kumar, Assistant Professor, Dept. of CSE, Methodist College of Engineering. & Technology.
Third Author – P. Vijaya Babu Varma, Assistant Professor, Dept. of CSE, Methodist College of Engineering & Technology.