

Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on Security

*An Oracle White Paper
February 2002*

Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on Security

EXECUTIVE OVERVIEW	3
INTRODUCTION.....	3
FUNDAMENTAL APPROACHES TO SECURITY.....	4
Why IBM's Approach Hurts Customers.....	4
Why Oracle's Approach Benefits Customers.....	5
Impact on Customers	5
DB2 is More Than A Single Database Product	6
Packaging and Development.....	7
STATE OF SECURITY IN ORACLE AND DB2	8
Assurance.....	8
Locus of Security.....	9
Platform Support.....	10
Cost of Ownership.....	10
DETAILED FEATURES COMPARISON.....	11
User Authentication.....	11
Strong Authentication.....	12
Authorization and Access Control.....	13
Privileges	13
Views for Access Control	13
Granular Access Control.....	15
Virtual Private Database.....	15
Label-Based Access Control.....	16
RACF	16
Encryption	17
Encryption in the Database	17
Network Encryption.....	18
LDAP Integration for Centralized User Management.....	19
Auditing.....	19
Fine-grained Auditing.....	20
SecureWay Auditing	21
Feature Summary.....	21
CONCLUSION.....	24

Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on Security

EXECUTIVE OVERVIEW

Security is a top concern for IT Managers, CIO's, and now CEO's, because a company's reputation is at stake if it does not judiciously protect its systems and the customer information the systems hold. Because an organization's data is largely stored in databases, executive management is growing inquisitive about the security offered by database vendors.

INTRODUCTION

With security at the top of the list of buyers' software concerns, people are thinking about security more than ever. Relational databases hold a significant portion of data stored in software, therefore today's database purchase decisions revolve around how secure the product is. Two leading relational database management system (RDBMS) vendors, Oracle and IBM, provide security solutions within their product lines.

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. 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. The paper is divided into three sections: a view of the overall security in Oracle database versus that in DB2 and related products, a slightly lower-level comparison of the state of Oracle security against IBM security, and lastly, a comprehensive feature comparison of the two product lines.

FUNDAMENTAL APPROACHES TO SECURITY

IBM Corporation and Oracle Corporation institute vastly different business models when it comes to security. They vary in how highly each holds security as an objective and how each implements it in their products.

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. 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. 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. IBM's model compels customers to pay to secure the products they purchase.

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.

Why IBM's Approach Hurts 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.

IBM's security business is solid. The security model they choose to secure the database, however, has flaws that impact 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.
- Higher up-front costs because of the additional products necessary to secure DB2. Customers must purchase a database that includes little out-of-the-box 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.

Why Oracle's Approach Benefits 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.

IBM provides security through services and applications, whereas Oracle provides security through the software.

Impact on Customers

The following table summarizes the impact on customers of the two companies' divergent approaches.

Table 1: Impact 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.

DB2 is More Than A Single Database Product

DB2 is actually three distinct products with three separate code bases: OS/390, AS/400 and Unix/NT/Linux. The Oracle database is one product family built on one code base. From a security perspective, IBM’s approach results in security interoperability problems across platforms, whereas Oracle’s approach results in the same, interoperable security solutions across operating systems.

Oracle’s methodology leverages the same DBA skill set across heterogeneous platforms, while IBM’s methodology means that DBAs who are expert at managing and securing the database on one platform cannot easily leverage their knowledge on other operating systems. The platform-to-platform differences are amplified when you consider the security impact. For example, DB2 on a mainframe takes advantage of Resource Access Control Facility (RACF) for access control and other security features, but the absence of RACF on the Unix/NT/Linux DB2 product means that administrators cannot secure all instances of DB2 without application developers re-coding applications and/or without purchasing additional products. With IBM, it is difficult to tell how much security you get; it depends on the product, the version, the platform, the add-on products, the consultants, and so on.

On the other hand, a customer can implement Oracle solutions, such as row-level access control, across any of the 30-plus platforms that support Oracle. Customers running Oracle on Windows, HP-UX, AIX, and Tru64 do not require DBAs to re-code the security implementation for each operating system. For example, take a Virtual Private Database (VPD) implementation that restricts access to rows only in a user’s department. A DBA can apply the same VPD policy to tables in databases on any (or all) operating systems. IBM cannot make the same claim. The results of Oracle’s methodology are: support

Oracle’s methodology leverages the same DBA skill set across heterogeneous platforms, while IBM’s methodology means that DBAs who are expert at managing and securing the database on one platform cannot easily leverage their knowledge on other operating systems.

for the same DBA skill set on any operating system, and consistent security enforcement on all platforms.

Packaging and Development

At first glance, one might draw the conclusion that Oracle and IBM follow the same packaging and development principles. IBM builds some security features into the database and others into an affiliated product set (e.g., Tivoli SecureWay). Likewise, Oracle builds some security features into the database and others into associated options (e.g., Oracle9i Advanced Security and Oracle9i Label Security). Why is this assumption incorrect?

The difference lies in the fact that IBM separates database security features from application security features at the *enterprise* level. Oracle separates database security features from option features primarily by *packaging*.

IBM runs two independent businesses (as described later); one business unit researches and develops the relational databases, and another develops security solutions. The two independent organizations operate under separate management structures, ship product on their own autonomous release schedules, and endeavor to meet distinctly different customer requirements. The DB2 group builds databases and Tivoli builds security.

The development business at Oracle Corporation represents a striking contrast. Oracle has a core security group that drives security for the entire company. At Oracle, the security development group *is* the database development group. Developers code the latest and greatest security mechanisms sitting in cubicles right next to developers working on National Language Support (NLS) character sets and report to the same management responsible for high availability. They work towards the same objectives from management, follow the same coding standards, and work together to produce integrated features.

Therefore, whether an Oracle security feature is part of Oracle9i Database Enterprise Edition or Oracle9i Advanced Security is primarily a packaging decision. This approach promotes three positive results: easier installation and configuration (the Oracle9i Advanced Security option installs transparently in database “typical install” mode), better ease-of-use, and, most importantly, tightly integrated security. A straightforward, integrated security model is the best and most secure way to build software. With all of the combinations and permutations of operating system security features, DB2 and Tivoli product lines, what assurance do you have that all features work together, much less work together securely? As Bruce Schneier, one of the foremost security experts has said, “As a security professional, I think [complexity] is terrifying. Complexity is the worse enemy of security.”¹

IBM has one business unit that develops relational databases, and another that develops security solutions. The two independent organizations operate under separate management structures, ship product on their own autonomous release schedules, and endeavor to meet distinctly different customer requirements.

At Oracle, the security development group is the database development group. They work towards the same objectives from management, follow the same coding standards, and work together to produce integrated features.

¹ Bruce Schneier, “Software Complexity and Security,” *Crypto-Gram Newsletter*, 0003, March 15, 2000.

STATE OF SECURITY IN ORACLE AND DB2

Now that we've established the differing ways in which Oracle and IBM address security, let's examine where the antithetical approaches position the two companies today.

Assurance

Any vendor can claim to build a secure product, but what *assurance* of a product's security does one have? There is no equivalent of a TPC benchmark for security, and with the database battles heating up, customers want a clear answer to the conflicting security claims they hear from competing companies. How can you be confident about the security built into a product? Independent security evaluations against internationally established security criteria provide assurance of vendors' security claims.

The Oracle9i database builds upon 14 independent security evaluations of its server software. Nine of those evaluations have examined the security of the Oracle database, and the first was completed eight years ago, in 1994.² The evaluation process, from inception to certificate, often lasts up to a full year (and sometimes longer); it is not a trivial task. Security evaluations are carried out by independent, licensed and accredited organizations. The evaluators not only examine the software design and code, but they also consider process aspects such as coding standards, development and production practices. Organizations who have undergone evaluations learn to improve upon their coding, testing and shipping processes as a result of completing the demanding process. No other database vendor approaches the number of evaluations that Oracle has, nor can they claim the years of experience from the efforts behind these evaluations.

IBM has not completed *any* evaluations of DB2. They therefore can claim little assurance of the security implementations in the product. Security evaluations are perhaps the most effective way to qualify a vendor's assertions about its security implementations because such evaluations provide independent evidence of properly implemented security against established criteria. Without completing any evaluations, IBM cannot truly back its DB2 security claims. It leads commercial customers to wonder how secure DB2 is, and it leaves government customers wondering how they could purchase DB2 at all. The fact that Oracle has dominant market share in U.S. Federal accounts is evidence that security-conscious customers choose Oracle. Is DB2 secure enough to

² The Oracle RDBMS has undergone and completed the following evaluations:
Common Criteria - Three Oracle RDBMS evaluations completed at level EAL4
ITSEC - Three Oracle RDBMS evaluations completed at level E3/F-C2
TCSEC - One Oracle RDBMS evaluation completed at C2 level
Russian - One Oracle RDBMS evaluation completed at level IV
Russian - One Oracle RDBMS evaluation completed at level III

There is no equivalent of a TPC benchmark for security, and with the database battles heating up, customers want a clear answer to the conflicting security claims from competing companies.

Independent security evaluations against internationally established security criteria provide assurance of vendors' security claims.

Oracle has completed nine evaluations of the database server, but IBM has completed no evaluations of DB2.

run mission critical applications? Is it secure enough to store their most sensitive data?

The answer for systems used in any U.S. national security is “no” because of new NSTISSP #11 government regulations. This is the National Security Telecommunications Systems Security Policy number 11 (in full effect in July 2002), which essentially states that *any* system involved in national security requires independent measures of assurance, such as a FIPS-140 certification or a Common Criteria (CC) evaluation. The U.S. Federal government *requires* these measures of assurance in their most sensitive applications and it is unlikely that there will be waivers granted by the National Security Agency to non-compliant products. It is unclear how Federal accounts could deploy DB2 when NSTISSP goes in effect this July.

Locus of Security

An oft-heard analogy in the security business is that implementing security outside of the product you’re trying to secure is like a bank locking the front door, but not the vault inside. Just as it is difficult to adequately protect the valuables in the vault without locking the vault itself, it is difficult to design software to adequately secure valuable information without locking down the database itself. Because the database holds the “crown jewels” of an organization’s data, it is vastly important to protect this repository.

IBM builds almost no security into DB2, relying largely upon the independently-developed and separately-sold Tivoli product line or the OS to deliver security solutions. As an article in *InfoWorld* states, “Tivoli Systems has re-branded IBM SecureWay and Tivoli security management offerings with the Tivoli SecureWay nameplate, but despite the apparent unification of the IBM and Tivoli product lines, a Tivoli official said the move reaffirms—rather than undermines—Tivoli’s autonomous status.”³ Keeping the security development organization separate and autonomous from the database development organization (and relying on Tivoli for all things security) means that DB2 is like the unlocked vault behind the bank’s locked door.

Oracle, conversely, protects the data itself where it is stored—in the database. Oracle provides a plethora of security features, from privilege management to row-level access control, something that no other vendor provides even in their add-on security products. Oracle also ships security options and partners with security companies, ostensibly adding a front door lock, an alarm system, and guard dogs to the bolted vault inside the bank.

Oracle protects the data itself where it is stored—in the database.

³ Paul Krill, “Tivoli, IBM Security Products Unified,” *InfoWorld*, January 21, 2000.

Platform Support

Earlier sections of the paper explained the inefficiencies and security issues resulting from platform-specific versions of DB2. Adding to those problems is the issue of securing the various DB2 products on multiple platforms. When IBM shipped DB2 version 7.2 on Linux in June 2001, they heavily promoted DB2 on Linux. However, they overlooked a crucial gap. The most important Tivoli product for securing DB2, SecureWay Policy Manager, was not available on Linux. IBM advertised DB2 for Linux regardless, overlooking the fact that customers could not secure it.

This scenario also illustrates the divergent production goals of the two organizations, Tivoli and the DB2 groups. The DB2 groups build databases and Tivoli builds security. Without Tivoli, there is little DB2 security. Oracle Corporation's database group *is* Oracle's security group. Customers enjoy the benefits of secure Oracle products from day one of General Availability.

Cost of Ownership

IBM and Oracle are going head-to-head in the cost of ownership debate. In order to compare a secured DB2 database to a secured Oracle database, one must add to the IBM TCO the cost of the Tivoli SecureWay product line. The consulting services often required to integrate the pieces add additional cost.

IBM forces customers to purchase the DB2 database, then add on the appropriate Tivoli SecureWay products for the customer's requirements. Additionally, customers oftentimes pay for IBM Global Services to integrate security in DB2 for one operating system that supplies a particular security mechanism, DB2 for another that doesn't natively support that mechanism, and any SecureWay pieces they choose. The choices are so complex that IBM actually has services called "IBM's Secure Product Selection."⁴ IBM charges customers to help them navigate through the complex security offerings.

Adhering to IBM's business model of making money on services, they use a piecemeal approach where customers pay for IBM services. Even if they price DB2 and SecureWay low, there is a high cost of integrating the products. Look no further than IBM's extensive list of security services to prove this point.⁵ This a fine business practice and a lucrative business model that may please IBM shareholders, but it results in IBM customers paying more to securely run their own businesses.

The security built into the Oracle database keeps the cost of ownership low for customers. There are no hidden charges for additional required products, nor for required consulting services. Oracle does, of course, offer supplementary

The choices are so complex that IBM has services called "IBM's Secure Product Selection."

⁴ See <http://www-1.ibm.com/services/security/pesspec.html>

⁵ See <http://www-1.ibm.com/services/security/index.html>

security options, such as Oracle Advanced Security, but even when you factor in those licensing costs, the Oracle solution is less expensive than IBM's.

State of Security in Oracle9i and IBM DB2

In summary, the difference in approach to security between IBM and Oracle has ripple effects throughout many areas. Between Oracle's far lead in independent security evaluations, their philosophy of securing the data itself, a consistent product across platforms, and lower cost of ownership, IBM has some catching up to do.

Table 3: State of Security

	Oracle9i Database	IBM DB2
Assurance	Oracle9i Database builds on 14 independent security evaluations. The evaluations substantiate Oracle's security claims.	IBM has not completed any independent security evaluations of DB2. No way to substantiate DB2 security claims.
Locus of Security	Security built into the database, where data resides.	Relies on applications (e.g., Tivoli SecureWay) or operating system for security.
Platform Support	Consistent product on all platforms, with security built-in from day one.	Lack of security features on many platforms. SecureWay ships on a different schedule, resulting in void in secure DB2 availability.
Cost of Ownership	Lower total cost of ownership.	Increased total cost of ownership.

DETAILED FEATURES 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.

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.⁶

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.

Strong Authentication

Authentication is used to prove the identity of the user, and, as discussed above, passwords are the most common means of authentication. Today there are a number of software services and hardware mechanisms that provide strong authentication, sometimes defined as “anything stronger than a password.” Strong authentication can involve network services, including MIT’s Kerberos and the Internet standard Remote Authentication Dial-In User Service (RADIUS). Two-factor authentication—proving user identity based on something the user has (e.g., a smart card) and something she knows (a personal identification number or PIN)—is another popular means of strongly authenticating users.

Oracle supports strong authentication at the database and network layers by supporting X.509v3 digital certificates and integrating with third-party network authentication services (including Kerberos, DCE and CyberSafe), token cards, biometrics, and smart cards. The RADIUS implementation in Oracle9i Advanced Security enables any RADIUS-compliant device to authenticate Oracle users—and it represents a transition from the former Oracle7/Oracle8 method of supporting only the best-of-breed token, best-of-breed biometric device, and so on. The RADIUS interface supports authentication to Oracle via SecurID tokens, Secure Computing SafeWord tokens and smart cards, and ActivCard tokens and smart cards, to name a few. It is a matter of packaging, but the Oracle database itself supports some of these services such as X.509v3 in some cases, and the Oracle9i Advanced Security option can be licensed for many of the strong authentication services in other cases.

⁶ *IBM® DB2® Universal Database Administration Guide: Implementation*

⁷ Most strong authentication mechanisms are packaged with the Oracle Advanced Security option.

IBM supports strong authentication at the database and operating system layers and in various Tivoli applications. At the database and operating system levels, IBM supports services such as DCE, Kerberos, and RACF (on mainframes). In terms of token cards, Tivoli SecureWay Policy Director supports only SecurID—the leading token, but leaving customers with only one choice. This model is akin to Oracle’s implementation circa 1997 and earlier when it supported only one best-of-breed token, biometric device and so on. However, IBM is ahead of Oracle with one authentication method: SecureWay supports industry-standard PKCS#11 smart cards. Supporting PKCS#11 means that any industry-standard smart card easily integrates with SecureWay. Additionally, SecureWay uses X.509v3 certificates for strong authentication over SSL, and it relies on IBM hardware on the client-side for those authentication services provided by the hardware.

Both IBM and Oracle deliver comprehensive strong authentication support.

In summary, as long as the customer is willing to spend additional licensing dollars, both IBM and Oracle deliver comprehensive strong authentication support.

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. 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* (Oracle term) or *authority levels* (DB2 term).⁸

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

⁸ IBM® DB2® *Universal Database Administration Guide: Implementation*

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 20.

Both DB2 and Oracle support the use of views to limit access to data.

Limitations of Views

While views can provide fairly granular access control, they have limitations which make them less than optimal for very granular access control:

Views can provide fairly granular access control, but they have limitations which make them less than optimal for very granular access control.

- Views are not always practical when you need a lot of them to enforce your security policy. For example, using views to restrict access to customer data by region is feasible if there are 10 customer regions (and hence 10 views). But it is not practical to limit customers' access to their own records if there are 100,000 customers (and hence 100,000 views).
- Views are best suited to access control conditions the database can evaluate simply. For example, you can create a view of the EMP table for employees who are in department 20 and whose salaries are less than \$50,000 if department and salary are columns in the table, and the database can evaluate the condition "less than 50,000." A more complex access control policy—or one in which the database cannot evaluate the access control condition—does not lend itself to views. Take, for example, an access control policy "a user accessing the EMP table as a Payroll clerk through the Payroll application is allowed to see all EMP information, including SALARY, but only for employees in her division." This is probably not possible to express in a view, since you can't determine what application the user is accessing at the time you create the view.
- If users access base tables, they bypass view security. While applications may incorporate and enforce security through views, users often need access to base tables to run reports or conduct ad-hoc queries. Users who have privileges on base tables are able to bypass the security enforcement provided by views. Note that this is a general problem of embedding security in applications instead of enforcing security through database mechanisms, but it is exacerbated when security is enforced on views and not on the data itself (that is, on the table containing the data).
- Views may complicate administration of security policy. A security administrator cannot tell the difference between the parts of a view definition based on logical object definition, and those designed to enforce security. When a security policy is added, changed, or removed, it's difficult to determine what exactly to do with each view. An

administrator cannot tell whether, by changing security policies through altering or dropping a view, she is breaking an application.

Due to the limitations of existing access control mechanisms, Oracle Corporation has developed a solution for a scalable, secure and lightweight means of limiting data access.

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.

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.

Many Oracle customers, representing a vast number of industries, use Virtual Private Database technology to separate data by customer, by organizational unit, geographical region, and so forth. Many use it because it lowers the cost of ownership. Customers enjoy the benefits of building security once, in the database, and certifying the core security code in the database, not multiple applications. Examples of VPD customers include:

- Several large banks and financial services companies use it to separate customer or employee access to financial data.
- An Application Service Provider (ASP) saves millions of dollars on hardware, DBAs and software because VPD enables it to host multiple customers' data in one database—with assurance of full data separation by customer number.
- Security-conscious U.S. Federal government organizations use it for even the most rigid implementations.
- A foreign government organization uses it in a large data warehouse.

Oracle's Virtual Private Database feature limits access to data at the row level and ties the security policy to the table itself.

Granular access control highlights the contrast between Oracle's model of building advanced features into the database engine and IBM's need to involve services to provide a commensurate solution.

- 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. [The customer] also developed custom code enabling administrators to permit or deny access - based on the user's privilege level...."⁹ Granular access control highlights the contrast between Oracle's model of building advanced features into the database engine and IBM's need to involve services to provide a commensurate solution.

Label-Based Access Control

Built on top of VPD, Oracle Label Security enforces label-based access control. Oracle9i Label Security is a security option for the Oracle Database that mediates access to data by comparing a sensitivity label on a piece of data with label authorizations assigned to an application user. Such access mediation allows data to be separated into different sensitivities within a single database.

Labels are used extensively in commercial and government organizations. Examples of labels include: internal, confidential, sensitive:human resources, and internal:Acme California. Oracle Label Security uses an Oracle-supplied security package to mediate access to data rows, and no coding or PL/SQL software development is required.

Again, IBM has no comparable solution. Label-based access control places Oracles years ahead of its competition in this area. Furthermore, Oracle Label Security (v8.1.7) is currently in evaluation against the Common Criteria (CC) at EAL4; completion of the evaluation will provide further assurance of this solid security solution.

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.

⁹ Tivoli SecureWay Policy Director Backgrounder

RACF on the mainframe augments Oracle's internal database security because Oracle supports RACF for customers running the Oracle database on mainframes.

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.

Encryption in the Database

Highly-publicized compromises of credit card numbers and personally identifiable information has prompted many organizations to consider encrypting especially sensitive data held in databases. Above and beyond other security mechanisms, one can obtain an additional measure of security by selectively encrypting sensitive data before storage 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.

The IBM solution enables the user to supply a password as the encryption key to encrypt and decrypt data. However, there has been no independent certification of implementation (e.g., FIPS), and there's nothing to stop a user from never changing a weak password which may be susceptible to a dictionary attack.

¹⁰ IBM® DB2® *Universal Database SQL Reference*

The Oracle solution is implemented securely, uses the FIPS-certified random number generator and runs in an evaluated database.

Each implementation has its advantages. IBM's password-based key provides flexibility if not a slight overburden on the end user to choose a strong key. DB2 is in its 1.0 release, where Oracle has made stored data encryption enhancements in four development cycles. And customers can be assured that the Oracle solution is implemented securely, as it uses the FIPS-certified random number generator and runs in an evaluated database. While both databases encrypt data, the more mature and certified Oracle implementation places Oracle ahead of DB2 in this area.

Network Encryption

Customers today demand a means of encrypting data passing over a network. For these customers, 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.

Wherever the Oracle database runs, the network traffic can be protected with encryption.

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), 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.

LDAP Integration for Centralized User Management

Among other vendors, Oracle and IBM are turning to Lightweight Directory Access Control (LDAP) directories to centrally store and manage users. Tivoli SecureWay User Administration provides an LDAPv3-compliant directory service for this purpose, and IBM supports LDAP on many operating systems. Oracle offers an LDAPv3-compliant directory service, Oracle Internet Directory, and many Oracle products use it as a scalable, secure central information repository. Both IBM and Oracle are involved in the LDAP standards committees.

Various IBM products employ one of the IBM LDAP directories. Specifically, IBM supports LDAP on OS/400, AIX, OS/390, NT and Windows, and these directories use DB2 to store directory information and are compatible with one another. Many IBM products use LDAP to authenticate users, access user information, manage product configurations, and the like.¹¹ Similarly, the Oracle directory product combines the flexibility of the Internet's LDAP v3 standard with the robustness of the Oracle database (Oracle Internet Directory runs on top of its own Oracle database) to provide a scalable, reliable and secure LDAP directory service. The Oracle Database—and other products including Oracle9i Application Server, Oracle Portal, Oracle Net Services, Oracle Email Server—harness the power of this directory to centrally administer users *and* to integrate with third-party LDAP directories.

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.

The DB2 audit facility produces an audit trail to capture database-level and instance-level events, and the implementation separates the audit facility from the DB2 instance in order to audit events that impact the DB2 instance itself. The database also provides an administrative tool called db2audit for use by the administrator responsible for auditing. A variety of auditing options are available, from auditing activities during authorization checking to auditing successful and unsuccessful attempts to access a particular objects. A highlight of the DB2 auditing facility is the ability to generate audit records when operations are performed by administrators. Finally, the database administrator has the option of configuring DB2 to audit synchronously or asynchronously.

¹¹ Directory Services (LDAP): What's new? Found at: <http://www-1.ibm.com/servers/eserver/series/ldap/ldapv4r5.htm>

The former means that the audited event does not execute until the record is written to disk, which ensures that all auditable events are captured but negatively impacts performance. The latter, asynchronous mode, uses a buffer to hold audit records before writing them to disk, so records can potentially be lost but it does not produce the same database performance issues.¹²

The Oracle audit facility allows customers to audit database activity by statement, by use of system privilege, by object, or by user—whether the operation is successful or unsuccessful. Audit trail records can be stored in a database table, making the information available for viewing through ad hoc queries or any appropriate application or tool, or combined with operating system audit trails on selected operating systems, for ease of management. Oracle implements auditing efficiently; statements are parsed once for both execution and auditing. Additionally, Oracle makes use of database logs to capture operations performed by administrators and every other user. Oracle captures all changes to the database, and they can be queried using the LogMiner utility. Thus, customers get the benefit of auditing without any additional overhead. Since the database must be recoverable, the logs are always available; Oracle does not drop records of any changes made to it. Auditing is implemented within the server itself, with a variety of audit options, allowing customers to record specific database activity without incurring the performance overhead that more general auditing entails.

In general, if not done carefully, the sheer volume of audit logs can make finding suspicious activities like searching for a needle in a haystack. Auditors and security administrators aim to reduce the amount of data logged but capture all relevant data. Granular auditing dramatically reduces the amount of data captured and hones in on the sensitive data that must be audited. Oracle9i Database expands the above auditing facilities and institutes fine-grained auditing.

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.

The Oracle event handler sets a triggering audit event that could activate a pager for the security administrator. DB2 offers no support for such granular and customizable auditing.

¹² *IBM® DB2® Universal Database Administration Guide: Implementation*

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

At least two Tivoli products, namely SecureWay Security Manager and SecureWay PKI, 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.

Feature Summary

The bottom line is that DB2 offers only fundamental database security mechanisms, while Oracle provides the same basic security features *along with* a host of mature, industry-leading security solutions. Both DB2 and Oracle support basic tasks like creating users, assigning passwords, and setting authorizations. Oracle uniquely goes on to build advanced features that allay

¹³ *Tivoli SecureWay Security Manager User's Guide V3.7*

customers’ concerns about the threats their databases face from hackers, disgruntled employees, and simple mismanagement of data. These advanced features—including row-level security, fine-grained auditing, encryption in the database—place Oracle many years ahead of DB2 and set IBM in a “catch up” position in the database space. The following table summarizes security features available in the Enterprise Edition of Oracle9i Database and the Enterprise Edition of DB2.

Table 4: Database Security Features

Feature or Area	Oracle9i Database Enterprise Edition	DB2 Enterprise Edition
Authorization	Yes	Yes
Auditing	Yes	Yes
Fine-grained Auditing	Yes	No
Stored Data Encryption	Yes	Yes
Fine-grained Access Control	Yes	No
LDAP Support	Yes	Yes
Proxy Authentication	Yes	No
PKI Support	Yes	No
Strong Authentication	Yes	Yes
Evaluated RDBMS	Yes (9 of RDBMS)	No (0 of DB2 RDBMS)

The chart above shows a high-level database comparison without looking at the implementation of the features, nor the maturity or completeness of the solution. Since IBM delivers security solutions in its operating systems and in Tivoli products, one might think that comparing the “whole nine yards” of IBM security solutions to those of Oracle would produce closer results.

However, even when you take into account the entire IBM security stack—from the operating system to the database to the application layer—they still do not measure up to the completeness of Oracle security. The following table takes all of the layers into account and shows a more complete picture of the robustness of the solutions. The comparison does *not* take into account the price of IBM Global Services required to integrate the pieces of the stack, the ineffectiveness of retraining DBAs to administer the same security on different platforms, nor the additional cost of the Tivoli products themselves. Moreover, there is no way to measure for certain the cost of building security outside of the database and the risk of users bypassing application-based security.

Table 5: Database, Options, OS, Tivoli Security Feature Comparison

Feature or Area	Oracle9i Database EE and options	IBM DB2 EE	Tivoli SecureWay or OS
Authorization	Yes	Yes	Yes

LDAP Support	Yes	Yes	Yes
Stored Data Encryption	Yes	Yes	N/A
Password Encryption Key	No	Yes	No
Fine-grained Access Control	Yes	No	No
Label-based Security	Yes	No	No
RACF Support	Yes (on mainframe)	Yes	Yes
Auditing, Basic Audit Tools	Yes	Yes	Yes
Fine-grained Auditing	Yes	No	No
Granular Audit and Log GUI	Yes	No	No
Proxy Authentication	Yes	No	No
Network Encryption	Yes	No	Yes
PKI Support	Yes	No	Yes
Centralized User Management in LDAP	Yes	Yes	Yes
Strong Authentication	Yes	Yes	Yes
Kerberos Support	Yes	Yes	Yes
DCE Support	Yes	Yes	Yes
RADIUS Support	Yes	No	No
Token Cards	Yes (VARIOUS RADIUS-COMPLIANT)	No	Yes (SECURID ONLY)
Smart Cards	Yes (VARIOUS RADIUS-COMPLIANT)	No	Yes (ANY PKCS#11)
Single Sign-On	Yes (DCE, KERBEROS, SSL/LDAP)	Yes (DCE, KERBEROS)	Yes
Evaluated RDBMS	Yes (9 of RDBMS)	No (0 of DB2 RDBMS)	Yes (OPERATING SYSTEM)

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. Even taking into account the security features in the Tivoli SecureWay product line, Oracle still beats IBM.

It is difficult to make up for a lack of security built into the core DB2 product set, but IBM offers a variety of packaged service plans to do so. This model ultimately hurts customers but keeps IBM profitable with revenue from services—a lucrative business model. Oracle's security solutions are much less expensive than IBM's because customers do not have to pay for additional software and services. 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. IBM's security solutions are less secure than Oracle's because they rely on external solution and services to implement security they've neglected to build into DB2, which does not provide equivalently robust, mature security features that Oracle has been shipping for years.



Technical Comparison of Oracle Database and IBM DB2 UDB: Focus on Security
February 2002

Author: Kristy Browder
Contributing Author: Mary Ann Davidson

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
www.oracle.com

Oracle Corporation provides the software
that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.

Copyright © 2000 Oracle Corporation
All rights reserved.