Oracle9*i* Database Security for eBusiness

An Oracle White Paper
June 2001

Oracle9*i* Security for eBusiness

EXECUTIVE OVERVIEW

Information is the cornerstone of eBusiness. The Internet allows businesses to use information more effectively, by allowing customers, suppliers, employees, and partners to get access to the business information they need, when they need it. Customers can use the web to place orders which can be fulfilled more quickly and with less error, suppliers and fulfillment houses can be engaged as orders are placed, reducing or eliminating the need for inventory, and employees can obtain timely information about business operations. The Internet also makes possible new, innovative pricing mechanisms, such as online competitive bidding for suppliers, and online auctions for customers. These Internet-enabled services all translate to reduced cost: there is less overhead, greater economies of scale, and increased efficiency. eBusiness ’greatest promise is more timely, more valuable information accessible to more people, at reduced cost of information access.

eBusiness’ greatest promise is more timely, more valuable information accessible to more people, at reduced cost of information access.

The promise of eBusiness is offset by the security challenges associated with the disintermediation of data access — “cutting out the middleman” too often cuts out the information security the middleman provides — and the expansion of the user community from a small group of known, vetted users accessing data from the intranet, to thousands of users accessing data from the Internet. Application hosting providers and exchanges offer especially stringent — and sometimes contradictory — requirements of per user and per customer security, while allowing secure data sharing among communities of interest.

Oracle9*i* addresses the above eBusiness security challenges through:

- *Deep data protection*, ensuring well-formed, comprehensive security from client to application server to data server, as well as throughout the layers of an application
- *Internet-scale security*, which allows user and privilege management to scale to hundreds of thousands of users accessing data
- *Secure hosting and data exchange*, enabling economical, secure partitioning of data access by customer or by user, while supporting secure data sharing among communities of interest

THE NEEDS OF EBUSINESS SECURITY

While putting business systems on the Internet offers potentially unlimited opportunities for increasing efficiency and reducing cost, it also offers potentially unlimited risk. The Internet provides much greater access to data, and to more valuable data, not only to legitimate users, but also to hackers, disgruntled employees, criminals, and corporate spies.

Increased Data Access

The Internet provides much greater access to data, and to more valuable data, not only to legitimate users, but also to hackers, disgruntled employees, criminals and corporate spies.

One of the chief eBusiness benefits of the Internet is "disintermediation." The intermediate information processing steps which employees typically perform in "brick and mortar" businesses, such as typing in an order received over the phone or by mail, are removed from the eBusiness process. Users who are not employees and are thus outside the traditional corporate boundary, including customers, suppliers, and partners, can have direct and immediate online access to business information which pertains to them.

In a traditional office environment, any access to sensitive business information is through employees. Although employees are not always reliable, at least they are known, their access to sensitive data is limited by their job function, and access is enforced by physical and procedural controls. Employees who pass sensitive information outside the company contrary to policy may be subject to disciplinary action; the threat of punishment thus helps prevent unauthorized access.

Making business information accessible via the Internet vastly increases the number of users who may be able to access that information. When business is moved to the Internet, the environment is drastically changed. Companies may know little or nothing about the users (including, in many cases, employees) who are accessing their systems. Even if they know who their users are, it may be very difficult for companies to deter users from accessing information contrary to company policy. It is therefore important that companies manage access to sensitive information, and prevent unauthorized access to that information before it occurs.

Much More Valuable Data

EBusiness relies not only on making business information accessible outside the traditional company, it also depends on making the best, most up-to-date information available to users when they need it. For example, companies can streamline their operations and reduce overhead by allowing suppliers to have direct access to consolidated order information. This allows companies to reduce inventory by obtaining exactly what they need from suppliers when they need it. Companies can also take advantage of new pricing technology, such as online competitive bidding via exchanges, to obtain the best price from suppliers, or offer the best price to consumers.

Streamlining information flow through the business system allows users to obtain better information from the system. In the past, data from external partners, suppliers, or customers was often entered into the system through inefficient mechanisms that were prone to error and delay. For example, many companies accepted the bulk of their orders by phone, letter, or fax, and this information was typed in by clerks or sales people. Even when electronic data interchange mechanisms existed, they were typically proprietary and difficult to integrate with companies' internal data infrastructure. Now, businesses that allow other businesses and consumers to submit and receive business information directly through the Internet can expect to get more timely, accurate, and valuable information, at less expense than if traditional data channels were used.

The sheer size of the user communities which can access business systems via the Internet not only increases the risk to those systems, it also constrains the solutions which can be deployed to address that risk.

Formerly, when information was entered into a business system, it was often compartmentalized. Information maintained by each internal department, such as sales, manufacturing, distribution, and finance, was kept separate, and was often processed by physically separate and incompatible databases and applications — so-called "islands of information." This prevented businesses from taking full advantage of the information they already had, since it was difficult for different departments to exchange information when it was needed, or for executives to get the latest and most accurate "big picture" of the business. Companies have found that linking islands of information and consolidating them where possible, allows users to obtain better information, and to get more benefit from that information, which thus makes the information more valuable.

Improving the value of data available to legitimate users generally improves its value to intruders as well, increasing the potential rewards to be gained from unauthorized access to that data, and the potential damage that can be done to the business if the data were corrupted. In other words, the more effective an eBusiness system is, the greater the need to protect it against unauthorized access.

Large User Communities

The sheer size of the user communities which can access business systems via the Internet not only increases the risk to those systems, it also constrains the solutions which can be deployed to address that risk. The Internet creates challenges in terms of scaleability of security mechanisms, management of those mechanisms, and the need to make them standard and interoperable.

Scaleability

Security mechanisms for Internet-enabled systems must support much larger communities of users than systems which are not Internet-enabled. Whereas the largest traditional enterprise systems typically supported thousands of users, many Internet-enabled systems have millions of users.

Manageability

Traditional mechanisms for identifying users and managing their access, such as granting each user an account and password on each system he accesses, may not be practical in an Internet environment. It rapidly becomes too difficult and expensive for system administrators to manage separate accounts for each user on every system.

Interoperability

Unlike traditional enterprise systems, where a company owns and controls all components of the system, Internet-enabled eBusiness systems must exchange data with systems owned and controlled by others: customers, suppliers, partners, etc. Security mechanisms deployed in eBusiness systems must therefore be standards-based, flexible, and interoperable, to ensure that they work with others 'systems. They must support thin clients, and work in multi-tier architectures.

Hosted Systems and Exchanges

The principal security challenge of hosting is keeping data from different hosted user communities separate. The simplest way of doing this is to create physically separate systems for each hosted community. The disadvantage of this approach is that it requires a separate computer, with separately installed, managed, and configured software, for each hosted user community, providing little economies of scale to a hosting company. Mechanisms which allow multiple different user communities to share a single hardware and software instance, keep data for different user communities separate, and allow a single administrative interface for the hosting provider, can greatly reduce costs for the hosting service provider.

Oracle9*i* provides deep data protection through Virtual Private Database, Oracle Label Security, selective data encryption, and extensive auditing.

Exchanges have requirements for both data separation and data sharing. For example, an exchange may ensure that a supplier 's bid remains unviewable by other suppliers, yet allow all bids to be evaluated by the entity requesting the bid. Furthermore, exchanges may also support "communities of interest" in which groups of organizations can share data selectively, or work together to provide a joint bid, for example.

ORACLE9I DEEP DATA PROTECTION

Deploying eBusiness systems on the Internet increases risk. Among the best ways to mitigate security risk is to provide multiple layers of security mechanisms, so that failure of a single mechanism does not result in compromise of critical information. We refer to this concept as *deep data protection*; Oracle9*i* provides it through Virtual Private Database(VPD), Oracle Label Security, selective data encryption, and extensive auditing.

The Virtual Private Database enables, within a single database, per-user or per-customer data access with the assurance of physical data separation.

Virtual Private Database

Oracle8*i* set a new standard in database security with the introduction of Virtual Private Database (VPD), unique to Oracle: server-enforced, fine-grained access control, together with secure application context, enabling multiple customers and partners to have secure direct access to mission-critical data. The Virtual Private Database enables, within a single database, per-user or per-customer data access with the assurance of physical data separation. For Internet access, the Virtual Private Database can ensure that online banking customers see only their own orders. Web hosting companies can maintain multiple companies ’data in the same Oracle9*i* database, while allowing each company to see only its own data.

Within the enterprise, the Virtual Private Database results in lower cost of ownership in deploying applications. Security can be built once, in the data server, rather than in each application which access 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 an ad hoc query tool or new report writer. Virtual Private Database is key enabling technology for organizations building hosted, web-based applications, as well as for Oracle itself. Multiple Oracle applications, including Oracle SalesOnline and Oracle Portal, use VPD to enforce data separation for hosting.

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 function implementing 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. A secure application context enables access conditions to be based on virtually any attributes an application deems significant, such as organization, cost center, account number, or position. For example, an Web order entry system can enforce access based on customer number, and whether the user is a customer or a sales representative. In this way, customers can view their order status online (but only for their own orders), while sales representatives can view multiple orders, but only for the their own customers.

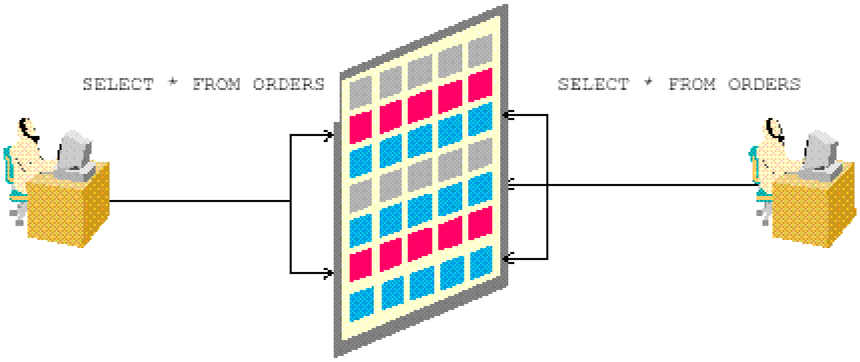

Figure 1: Virtual Private Database: Customers See Only Their Own Orders

The Virtual Private Database ensures that, no matter how a user gets to the data, the same strong access control policy is enforced.

The Virtual Private Database ensures that, no matter how a user gets to the data (through an application, a report writing tool, or SQL*Plus®) the same strong access control policy is enforced. The Virtual Private Database can help banks ensure that customers see their own accounts (and nobody else’s), that telecommunications firms can keep customer records safely segregated, and that human resources applications can support their complex rules of data access to employee records. The Virtual Private Database is a key enabling technology in building three-tier systems which expose mission-critical resources to customers and partners.

Oracle9*i* expands the Virtual Private Database by adding multiple new enhancements:

- Oracle Policy Manager, a tool to facilitate security policy administration
- partitioned fine-grained access control, to ease VPD deployment in multi-application and hosted environments
- global application context, to support application user models
- application context that can be initialized from an external source, such as Oracle Internet Directory

Oracle Policy Manager

Oracle Policy Manager provides an easy-to-use graphical user interface to administer both VPD policies and Oracle Label Security policies.

Oracle9*i* offers improved management of VPD policies through Oracle Policy Manager, an easy-to-use graphical user interface (GUI) accessed through Oracle Enterprise Manager. Developers can use Oracle Policy Manager to apply security policies to schema objects, such as tables and views, as well as creating application contexts, thus making VPD much easier to develop and manage. Oracle Policy Manager is also the administration tool for Oracle Label Security, a VPD-based product that provides label-based access to data.

Partitioned Fine-grained Access Control

Oracle9*i* provides enhanced ability to partition security policy enforcement by application, thus facilitating VPD deployment. For example, suppose both an Order Entry and Inventory application access the Orders table. The Order Entry application limits access based on customer number, while the Inventory application limits access based on part number. It is very useful to be able to "partition" fine-grained access control so that *different* security policies apply, depending upon which application is accessing the data. Otherwise, application developers of the respective Order Entry and Inventory applications have to agree upon a mutual policy, which may not be feasible or possible.

Oracle9*i* enables partitioning of Virtual Private Database through policy groups and a driving application context. A driving application context securely determines which application is accessing data, and policy groups facilitate managing the policies which apply by application. Oracle9*i* also supports default policy groups, which always apply to data access. For example, an application "striped" for application hosting using a subscriber ID could have a default policy, "Subscriber," that always enforces data separation by subscriber, and additional policy groups for Inventory and Order Entry-based access, which apply depending on the particular application accessing data.

Partitioned application context facilities application development using VPD, because development groups no longer need to collude; applications can have different security policies based upon their individual application needs.

Global Application Context

Many web-based applications use connection pooling to achieve high scalability and thereby support hundreds of thousands of users. These applications set up and reuse connections instead of having different sessions for each user. For example, web user Jane and Ajit connect to a middle tier application, which establishes a session in the database used by the application on behalf of both users. The application is responsible for switching the username on the connection, so that, at any given time, it's either Jane or Ajit using the session.

Global application contexts allow multiple sessions to share the same security attributes.

Oracle9*i* VPD capabilities facilitate connection pooling by allowing multiple connections to access one or more *global* application contexts, instead of setting up an application context for each user session. Global application contexts provide additional flexibility for web-based applications to use Virtual Private Database, as well as enhanced performance through reuse of common application contexts among multiple sessions instead of setting up per-session application contexts.

Application user proxy authentication can be used with global application context for additional flexibility and high performance in building eBusiness applications. For example, suppose a web-based application that provides information to business partners has three types of users: Gold, Silver, and Bronze, representing different levels of information available. Instead of each user having his own session — with individual application contexts — set up, the application could set up global application contexts for Gold, Silver or Bronze and use the client identifier to point the session at the correct context, in order to retrieve the appropriate type of data. The application need only initialize the three global contexts once, and use the client identifier to access the correct application context to limit data access. This provides performance improvements through session reuse, and through accessing global application contexts set up once, instead of having to initialize application contexts for each session individually.

Externalized Application Context

Many organizations centralize user and privilege management in a directory based on the Lightweight Directory Access Protocol (LDAP). Oracle9*i* also supports centralized management of users in Oracle Internet Directory, an LDAP-based directory built on the Oracle9*i* database (as described in Enterprise User Security, below).

In Oracle9*i*, VPD has been enhanced to easily populate application context attributes from attributes stored in Oracle Internet Directory. The ability to identify attributes in Oracle Internet Directory that can be used for initialization of an application context further enhances the ability of organizations to leverage directory-based user management and reap lower cost of ownership.

For example, an Order Entry application context could be initialized externally by populating "position," "cost center," and "region" attributes automatically, based on corresponding attributes defined for a user in Oracle Internet Directory. The ability to predefine "externally initialized" application contexts reduces the cost of development, since developers do not need to write LDAP calls to retrieve attributes from a directory into an application context. This also avoids duplication of data in both a database and a directory, by enabling VPD to use virtually *any* attribute stored in Oracle Internet Directory for fine-grained access control decisions.

Secure Application Role

A long-standing security problem has been that of limiting how users access data, to prevent users from bypassing application logic to access data directly. For example, in web-based applications, even if users are known to the database, it may not be desirable to allow them to have direct access to data. To-date, this has been a very difficult security problem to solve, because there has been no secure way to validate which application is used to access data — e.g. a malicious user could write a program that *appears* to be a valid human resources application, for example.

Oracle9*i* addresses this challenge through a secure application role: a role implemented by a package. The package can perform any desired validation to ensure that the appropriate conditions are met before the user can exercise privileges granted to the role in the database. The database ensures that it is only the trusted package implementing the role that determines the correct access conditions.

In three-tier systems using proxy authentication, the package can validate that the user session was created by a middle tier, and thus that the user is accessing the database through the correct application. The secure application role can also ensure that a user connecting directly to the database is not able to access any data. A secure application role can enforce other security conditions, as well; for example, the user may not be allowed to access especially sensitive human resources data from the Internet.

A secure application role enhances the native strong authentication and fine-grained access control of the database to prevent users from assuming any privileges unless the correct access conditions are met. Secure application role solves a very difficult security issue and supports secure web-based application data access.

Oracle Label Security

The ability to natively manage labeled data is a tremendous advantage to eBusinesses in being able to provide the right data to the right people at the right level of secure data access.

Oracle Label Security, a new security option for Oracle9*i*, extends Virtual Private Database to enforce label-based access control in the Oracle9*i* database. Oracle Label Security provides VPD "out-of-the-box," as well as automatic labeled data management, thus increasing the ease of deploying secure web-based eBusiness systems to customers, employees, and partners.

Label-based access control provided by Oracle Label Security allows organizations to assign sensitivity labels to information, control access to that data based on those labels, and ensure that data is marked with the appropriate sensitivity label. For example, an eBusiness may differentiate between "Company Confidential" information and "Public" information. Further, there may be some "Company Confidential" information that can be shared with partners, under a Confidential Disclosure Agreement or other legal document, and some that is only accessible by certain groups within the company, such as Finance or Sales divisions. The ability to *natively* manage labeled data is a tremendous advantage for eBusinesses in being able to provide the right information to the right people at the right level of secure data access.

Oracle Label Security Policies

Oracle Label Security policies are collections of labels, user authorizations and security enforcement options. Once created, policies can be applied to entire application schemas or specific application tables. Oracle Label Security supports multiple policy definitions within a single Oracle database. Label definitions, user authorizations and enforcement options are defined on a per policy basis. For example, a defense policy might have labels such as Secret, Top Secret and Confidential. A Human Resources policy might have labels such as HR-Only, Manager, and Senior VP.

Label Components

Oracle Label Security provides multi-dimensional, flexible data labeling capabilities. Oracle Label Security labels can include the following components:

Level — a hierarchical component which denotes the sensitivity of the data. A typical government organization might define levels confidential, sensitive and highly sensitive. However, there is no requirement to define more than one level. For example, a commercial organization might define a single level for company confidential data or application hosting requirements.

Compartment — a component, sometimes referred to as a category, that is non hierarchical. For example, a compartment might be defined for an ongoing strategic initiative or map to a hosted application subscriber. Oracle Label Security supports up to 9999 unique compartments.

Group — a component used to record ownership, that can be used hierarchically. For example, two groups called Senior VP and Manager could be created and subsequently assigned as children of the CEO group, creating an ownership tree.

Labels can be composed of a standalone level component, or a level component can be combined with compartments, groups or both.

Oracle Label Security Access Mediation

Oracle Label Security mediates access to rows in database tables based on a label contained in the row, a label associated with each database session, and Oracle Label Security privileges assigned to the session.

Oracle Label Security provides VPD enforcement "out-of-the-box," as well as automatic labeled data management, increasing the ease of deploying secure web-based eBusiness systems.

Oracle Label Security provides access mediation on an application table after a user has been granted the standard Oracle9*i* system and object privileges. For example, assume a user has SELECT privilege on an application table. If the user executes a SELECT statement on the table, Oracle Label Security will evaluate each row selected and determine if the user can access it based on the privileges and access labels assigned to the user by the security administrator. Oracle Label Security also performs security checks on UPDATE, DELETE, and INSERT statements. In addition, Oracle Label Security provides the ability to create trusted stored program units. These can be assigned privileges to perform operations outside a user's assigned label and privilege set.

Label Functions

Oracle Label Security offers flexibility in data labeling through label functions. Label functions can be defined in the Oracle database and referenced in an Oracle Label Security policy definition. Label functions compute the label value which should be assigned to application data during INSERT and UPDATE statements. Labeling functions can also draw upon the Virtual Private Database application context. For example, an application could (using the IP address accessed in a user's session) label the data differently depending upon whether the user is accessing data from the Intranet or the Internet. Label functions can be written in PL/SQL and assigned to Oracle Label Security policies through the Oracle Policy Manager (OPM) graphical user interface. Label functions are an extremely powerful feature of Oracle Label Security.

Selective Data Encryption

Among other security technologies, Oracle protects data in eBusiness systems through strong, standards-based encryption. Oracle has supported encryption of network data though Oracle Advanced Security since Oracle7. Oracle9*i* also supports protection of selected data via encryption within the database. Although encryption is not a substitute for effective access control, one can obtain an additional measure of security by selectively encrypting sensitive data before it is stored in the database. Examples of such data could include:

- credit card numbers

- national identity numbers
- passwords for applications whose users are not database users

Oracle9*i* provides secure stored data encryption using industry-standard DES and triple DES algorithms.

To address the need for selective data encryption, Oracle9*i* provides a PL/SQL package to encrypt and decrypt stored data. The package, DBMS_OBFUSCATION_TOOLKIT, supports bulk data encryption using the Data Encryption Standard (DES) algorithm, and includes procedures to encrypt and decrypt using DES. In addition to single DES, Oracle's DBMS_OBFUSCATION_TOOLKIT supports triple DES (3DES) encryption, in both two and three key modes, for those who demand the strongest commercial available level of encryption. The toolkit also supports the MD5 secure cryptographic hash to ensure data integrity, and a Federal Information Processing Standard (FIPS) 140-certified random number generator for generating secure encryption keys.

Auditing

A critical aspect of any security policy is maintaining a record of system activity to ensure that users are held accountable for their actions. Auditing helps deter unauthorized user behavior which may not otherwise be prevented. It is particularly useful for ensuring that authorized system users do not abuse their privileges. Oracle9*i* builds upon the existing robust and comprehensive auditing capabilities of the database to include fine-grained auditing, that can serve as an "early warning system" of users misusing data access privileges, as well as an intrusion detection system for the database itself.

Robust, Comprehensive Auditing

The Oracle9*i* audit facility allows businesses to audit database activity by statement, by use of system privilege, by object, or by user. For example, one can audit activity as general as all user connections to the database, and as specific as a particular user creating a table. One can also audit only successful operations, or unsuccessful operations. For example, auditing unsuccessful SELECT statements may catch users on "fishing expeditions" for data they are not privileged to see. Audit trail records can be stored in an Oracle9*i* 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.

Efficient Auditing

Oracle9*i* implements auditing efficiently: statements are parsed once for both execution and auditing, not separately. Also, auditing is implemented within the server itself, not in a separate, add-on server which may be remotely situated from the statements which are being executed (thereby incurring network overhead). The granularity and scope of these audit options allow Oracle customers to record and monitor specific database activity without incurring the performance overhead that more general auditing entails. And, by setting just the options of interest, Oracle9*i* customers can avoid "catch-all, and throw away" audit methods which intercept and log all statements, and then filter them to retrieve the ones of interest.

Customizable Auditing

To record customized information that is not automatically included in audit records, Oracle9*i* can use triggers to further customize auditing conditions and audit record contents. Database triggers are user-defined sets of PL/SQL or Java statements, stored in compiled form. While users explicitly execute stored procedures, database triggers are automatically executed (or "fired") within the data server based on pre-specified events. A trigger is defined to execute either before or after an INSERT, UPDATE or DELETE, so that when that operation is performed on that table, the trigger automatically fires. For example, one could define a trigger on the EMP table to generate an audit record whenever an employee's salary is increased by more than 10 percent and include selected information, such as before and after values of SALARY.

Fine-grained, Extensible Auditing

Oracle9*i* introduces extensible, fine-grained auditing, that can alert administrators to misuse of legitimate data access rights as well as serving as an intrusion detection system for the database.

Oracle9*i* expands upon the existing robust, granular auditing capabilities of the database by introducing extensible, fine-grained auditing. Fine-grained auditing enables organizations to define specific audit policies that can alert administrators to misuse of legitimate data access rights.

Fine-grained auditing allows organizations to define audit policies, which specify the data access conditions that trigger the audit event, and use a flexible event handler to notify administrators 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).

For additional flexibility in implementation, organizations can employ a user-defined function to determine the policy condition, and identify a relevant column for auditing ("audit column"). For example, the function could allow unaudited access to *any* salary as long as the user is accessing data within the intranet, but audit access to executive-level salaries when they are accessed from the Internet. An audit column helps reduce the instances of false or unnecessary audit records, because the audit need only be triggered when a particular column is referenced in the query. For example, an organization may only wish to audit executive salary access when an employee name is accessed, because accessing salary information alone is not meaningful unless an HR clerk also selects the corresponding employee name.

Oracle9*i* captures the exact SQL text of the statement the user executed in audit tables. In conjunction with other database features such as Flashback Query, fine-grained auditing can be used to recreate the exact records returned to a user. This may be especially important to organizations who have especially sensitive information they wish to share, for which they require strict accountability. For example, many law enforcement organizations at the international, federal, state and local level are increasingly becoming "eBusinesses" by sharing information among themselves, yet it is more important than ever that they audit access to sensitive information, such as informant data, to know who accessed what *exact* data.

The event handler provides organizations with flexibility in determining how to handle a triggering audit event. A triggering audit event could be written into a special audit table for further analysis, or could activate a pager for the security administrator. The event handler allows organizations to fine-tune their audit response to appropriate levels of escalation.

Fine-grained auditing enables organizations to hone their auditing capabilities to capture and identify particular, specific data access of concern. In addition to providing more granular, targeted audit information, such as detecting misuse of legitimate access, fine-grained auditing can also serve as an intrusion detection facility for the Oracle9*i* database itself.

Auditing For Three-Tier Applications

Many three-tier applications authenticate users to the middle tier, then the transaction processing monitor or application server connects as super-privileged user, and does all activity on behalf of all users. With Oracle9*i*, Oracle customers are not only able to preserve the identity of the real client over the middle tier and enforce "least privilege" through a middle tier, but can also audit actions taken on behalf of the user by the middle tier. Oracle9*i*'s audit records capture both the logged-in user (e.g., the middle tier) who initiated the connection, and the user on whose behalf an action is taken. Auditing user activity, whether users are connected through a middle tier or directly to the data server, enhances user accountability, and thus the overall security of multi-tier systems.

ORACLE9I INTERNET SCALE SECURITY

Security mechanisms must scale to Internet size — support many thousands or millions of users — and still be practical to administer. Oracle9*i* provides a number of security features tailored to building Internet-scale applications, including proxy authentication, support for Internet standards such as Secure Sockets Layer (SSL) and relevant Public Key Infrastructure (PKI) standards, Java security, and enterprise user management.

Proxy Authentication

Oracle9*i* provides a number of security features tailored to building Internet-scale applications, including proxy authentication, support for Internet standards such as SSL and relevant PKI standards, Java security, and enterprise user management.

Perhaps the most useful security feature in Oracle9*i* for supporting three-tier systems is the ability to proxy authenticated user identity from a middle tier to the database. The OCI proxy authentication feature was initially released in Oracle8*i*, and allowed a database client to set up, within a single database connection, a number of "lightweight" user sessions, each of which is associated with a different database user.

The feature is designed so that a specific middle tier can be restricted to acting on behalf of a specified set of users. Once the middle tier has authenticated itself to the database, it can establish a lightweight session on behalf of those users without submitting user-specific authentication information such as passwords. Moreover, Oracle9*i* can be configured so that a specific middle tier can assume a specific set of database roles when acting at the database on behalf of a specific user. In other words, the database uses both middle tier identity and client user identity when determining what privileges to grant a middle tier acting for a user through a lightweight session.

Oracle9*i*'s proxy authentication feature addresses a number of security problems associated with three-tier systems. Since each middle tier can be delegated ability to authenticate and act on behalf of a specific set of users, and with a specific set of roles, proxy authentication supports a limited trust model for the middle tier server, and avoids the problem of an all-privileged middle tier. It is also possible to give more privilege to a trusted middle tier (e.g., one that is within the corporate firewall) than to a less-trusted middle tier (e.g., one that is outside the firewall and thus more vulnerable to compromise). Moreover, because the identity of both middle tier and user are passed to the database through a lightweight user session, this feature makes it easier to audit the actions of users in a three-tier system, and thus improves accountability.

This feature has been enhanced in Oracle9*i*, to include:

- support for additional protocols
- expanded credential proxy
- application user proxy authentication

Support for Additional Protocols

In Oracle8*i* the proxy authentication feature was limited to communications to the database which used the Oracle Call Interface (OCI), but in Oracle9*i* the feature has been extended to "thick" Java Database Connectivity (JDBC) access to the database ("thick" vs. "thin" JDBC are discussed in the section of this paper on Java security). A middle tier server can now access the Oracle9*i* database on behalf of a client user by establishing a lightweight session for that user through either OCI or JDBC.

Expanded Credential Proxy

Oracle8*i* supported proxy authentication for database users authenticated by password only; the password could be passed as an attribute to be verified by the database, or not, depending on an organization's security preferences.

Oracle9*i* extends proxy authentication to include additional credential proxy of either the Distinguished Name (DN) or full X.509 certificate to the database.

Oracle9*i* extends proxy authentication to include additional credential proxy of either the Distinguished Name (DN) or full X.509 certificate to the database. This provides strong, three-tier security by enabling an SSL credential — an X.509 certificate or DN — to be passed to the database for purpose of identifying (but not authenticating) the user. (SSL cannot be used to authenticate a user through multiple tiers, since it is a point-to-point protocol rather than an end-to-end protocol.) For example, a user can authenticate to a middle tier using SSL, the middle tier can extract the DN from the certificate and pass it (or the full certificate) to the database. As an additional benefit, the DN or certificate is available in the lightweight session and the elements contained therein can be used with Virtual Private Database to limit access. For example, an organization could restrict data access based on the Organizational Unit (OU) element in a user certificate presented to the database.

The database can use the DN or certificate to look up a user in Oracle Internet Directory or other LDAP-based directory certified for Enterprise User Management (an Oracle Advanced Security feature). Integration of proxy authentication with Enterprise User Security enables the user identity to be maintained throughout all tiers of an application, yet the user need only be created once, in the directory. This also enables Enterprise User Security to be used in three-tier applications, instead of merely client-server, as was the case with Oracle8*i*.

Application User Proxy Authentication

Many applications use session pooling to set up a number of sessions which are reused by multiple users. In this context, "application users" are users who are authenticated to the middle tier of an application, but are not known to the database. Oracle9*i* introduces application user proxy authentication for these types of applications.

In this model, the middle tier passes a *client identifier* to the database upon session establishment. (The client identifier could be anything that represents the client connecting to the middle tier; a cookie, for example, or an IP address.) The client identifier, representing the application user, is available in user session information and can also be accessed within an application context (using the USERENV naming context), thus enabling applications to use Virtual Private Database to limit user access, even if the application users are not known to the database. Applications can set up and reuse sessions, while still being able to keep track of the "application user" in the session.

Applications can easily reset the client identifier and thus reuse the session for a different user, enabling high performance for web-based applications. For OCI-based connections, alteration of the client identifier is piggybacked on other OCI calls, to further enhance performance.

Application user proxy authentication is particularly valuable in eBusiness applications with thousands of users, as it supports per-user data access while meeting user scalability requirements.

Application user proxy authentication, available in thin JDBC, thick JDBC and OCI, provides the benefits of connection pooling without the overhead of setting up and managing separate user sessions (even "lightweight" ones), and enables even those applications whose users are unknown to the database to utilize Virtual Private Database. Application user proxy authentication is thus particularly valuable in eBusiness applications with thousands of users, as it supports per-user data access while meeting user scalability requirements.

SSL

Oracle9*i* implements the SSL protocol for encryption of data exchanged between database clients and the database. This includes data in Net8, LDAP, thick JDBC, and IIOP format. SSL encryption provides users with an alternative to the native Net8 encryption protocol which has been supported in Oracle Advanced Security (formerly known as Advanced Networking Option) since Oracle7. A benefit of SSL is that it is a de facto Internet standard, and can be used with clients which use protocols other than Net8.

In a three-tier system, SSL support in the database means that data exchanged between the middle tier and the database can be encrypted using SSL. The SSL protocol has gained confidence of users, and it is perhaps the most widely-deployed and well-understood encryption protocol in use today. Oracle9*i*'s implementation of SSL supports the three standard modes of authentication, including anonymous (Diffie-Hellman), server-only authentication using X.509 certificates, and mutual (client-server) authentication with X.509.

Oracle9*i* Application Server also supports SSL encryption between thin clients and the Oracle9*i* Application Server, as well as between Oracle9*i* Application Server and Oracle9*i* Data Server. As in Oracle9*i*, anonymous, server-only, and client-server authentication via X.509 are supported.

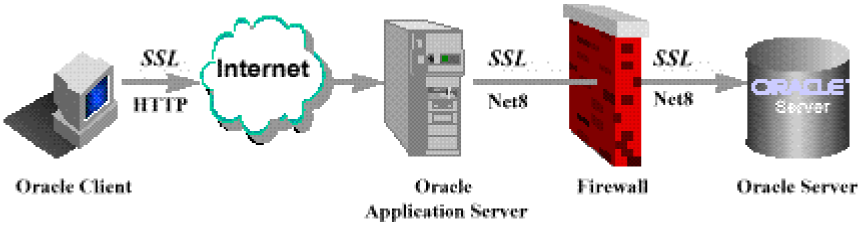

Figure 2: SSL Secures Internet and Oracle Communications

SSL addresses the problem of protecting user data exchanged between tiers in a three-tier system. By providing strong, standards-based encryption, SSL provides system developers and users with confidence that data will not be compromised in the Internet. Note also that unlike password-based authentication, which authenticates client to server only, SSL can authenticate server to client as well as client to server. This is a useful feature when building a web-based three-tier system, since users often insist on authenticating the identity of a web server before they will provide the server with sensitive information, such as credit card numbers.

Java Security

Oracle8*i* was the first relational database to provide built-in support for Java, reinforcing its position as the database platform of choice for Internet developers. The security model in Oracle8*i* is that of JDK 1.1, which provided relatively coarse-grained access control. Oracle9*i* extends this security model to that of JDK 1.2, which includes a fine-grained, policy-based access control model. This model is more flexible and configurable than the previous Java security model, and is based on a permission class hierarchy.

JDBC Security

JDBC is an industry-standard Java interface that provides a Java standard for connecting to a relational database from a Java program. Sun Microsystems defined the JDBC standard, and Oracle Corporation, as an individual provider, implements and extends the standard with its own JDBC drivers. Oracle implements two types of JDBC drivers: Thick JDBC drivers built on top of the C-based Net8 client, and thin (pure Java) JDBC drivers to support downloadable applets.

Since thick JDBC uses the full Net8 communications stack on both client and server, it can take advantage of existing Oracle Advanced Security encryption and authentication mechanisms. Because the thin JDBC driver is designed to be used with downloadable applets used over the Internet, Oracle9*i* includes a 100% Java implementation of Oracle Advanced Security encryption and integrity algorithms for use with thin clients. Oracle Advanced Security provides the following features for thin JDBC:

- Data encryption
- Data integrity checking
- Secure connections from thin JDBC clients to the Oracle9*i* database
- Ability for developers to build applets that transmit data over a secure communication channel
- Secure connections from Oracle9*i* databases to older versions of Oracle Advanced Security-enabled databases

Secure Connections for Virtually Any Client

Thick JDBC contains a complete implementation of a Net8 client in pure Java.

On the server, the negotiation of algorithms and the generation of keys function exactly the same as Oracle Advanced Security Net8 encryption, thus allowing backward and forward compatibility of clients and servers. On the clients, the algorithm negotiation and key generation occur in exactly the same manner as C-based Oracle Advanced Security encryption. The client and server negotiate encryption algorithms, generate random numbers, use Diffie-Hellman to exchange session keys, and use the Oracle Password Protocol, in the same manner as traditional Net8 clients. Thin JDBC contains a complete implementation of a Net8 client in pure Java. Consistent with other encryption implementations, the Java implementation of Oracle Advanced Security prevents access to the cryptographic algorithms, makes it impossible to double encrypt data, and encrypts data as it passes through the network. Users cannot alter the keyspace nor alter the encryption algorithms themselves.

Use of the Secure JDBC Implementation

The Oracle Advanced Security Java implementation gives developers the ability to build applets that transmit data over secure communication channels secured by Oracle Advanced Security. For example, it provides secure connections from any middle tier server with Java Server Pages (JSPs) to the Oracle9*i* Data Server and secure connections from Oracle9*i* databases to older versions of Oracle Advanced Security-enabled databases. This allows eBusinesses deploying Oracle and other components to securely transmit a variety of information over a variety of channels.

PKI Support

Oracle9*i* expands PKI integration and interoperability through PKCS#12 support, wallet storage in Oracle Internet Directory, multiple certificates per wallet, and strong wallet encryption.

Public Key Infrastructure (PKI) has emerged as the authentication technology which is most appropriate for securing Internet and e-commerce applications. There are a number of reasons for this. One is that PKI is highly scaleable. Since users maintain their own certificates, and certificate authentication involves exchange of data between client and server only (i.e., no third party authentication server needs to be online), there is no limit to the number of users which can be supported using PKI. Moreover, PKI allows delegated trust. That is, a user who has obtained a certificate from a recognized and trusted CA can authenticate himself to a server the very first time he connects to that server, without that user having previously been registered with the system.

As noted in the section on SSL, Oracle9*i* supports standard X.509 version 3 certificates and relevant Public Key Certificate Standards (PKCS) for certificate request and installation. This allows users to request certificates from any certificate authority (CA) which also supports these standards. It also allows users to install trusted root certificates from their choice of CAs, allowing the server to recognize and validate certificates issued by those CAs. Oracle is working with leading PKI service and product vendors, including VeriSign, Entrust, and Baltimore Technologies, to ensure that their CA trusted roots are pre-installed in

Oracle9*i*, allowing customers to use certificates from those vendors to authenticate to Oracle9*i* out-of-the-box.

Oracle9*i* expands PKI integration and interoperability through:

- PKCS#12 support
- wallet storage in Oracle Internet Directory
- multiple certificates per wallet
- strong wallet encryption

PKCS #12 Support

PKCS #12 support provides interoperability with third-party applications including browsers.

Oracle Advanced Security supports X.509 certificates stored in PKCS #12 containers, making the Oracle wallet interoperable with third party applications like Netscape Communicator 4.x and Microsoft Internet Explorer 5.x, and providing wallet portability across operating systems. Users who have existing PKI credentials may export them in PKCS#12 format and reuse them in Oracle Wallet Manager, and vice versa. PKCS#12 thus increases interoperability and reduces the cost of PKI deployment for organizations.

Wallets Stored in Oracle Internet Directory

Oracle Enterprise Security Manager creates user wallets as part of the user enrollment process. The wallet is stored in Oracle Internet Directory, or other LDAP-compliant directory. Oracle Wallet Manager can upload wallets to— and retrieve them from— the LDAP directory.

Storing the wallet in a centralized LDAP-compliant directory lets users access them from multiple locations.

Storing the wallet in a centralized LDAP-compliant directory supports user roaming, allowing users to access their credentials from multiple locations or devices, ensuring consistent and reliable user authentication, while providing centralized wallet management throughout the wallet life cycle.

Multiple Certificate Support

In Oracle9*i*, Oracle Wallet Manager and Oracle Enterprise Login Assistant support multiple certificates for each wallet, including:

- S/MIME signing certificate
- S/MIME encryption certificate
- Code-signing certificate

Oracle Wallet Manager Version 3.0 supports multiple certificates for a single digital entity in a persona— with multiple private key pairs in a persona (each private key can match only one certificate). This enables consolidation of and more secure management of users ’PKI credentials.

Strong Wallet Encryption

The private keys associated with X.509 certificates require strong encryption, over secure channels. Oracle9*i* replaces DES encryption with 3-key triple DES (3DES), which is a substantially stronger encryption algorithm and provides superior security for Oracle wallets.

Enterprise User Security

Most organizations, whether eBusinesses or not, face daunting obstacles in user management. Users within an organization often have far too many user accounts, a problem exacerbated by the growth in web-based self-service applications — every other week, users have a new user account and password to remember. Organizations who want "per user" data access and accountability do not want the administrative nightmare of managing users in each database a user accesses.

This problem is compounded for web-facing, eBusiness applications. An organization opening its mission-critical systems to partners and customers does not want to create an account for each partner in each database the partner accesses, yet "per partner" privilege and "per partner" accountability is highly desired. Oracle Advanced Security's enterprise user security feature, consisting both of enterprise privilege administration and of schema-independent users, addresses the requirement of per-user data access with centralized user management.

Enterprise Privilege Administration

An inherent challenge of any distributed system, including three-tier systems, is that common application information is often fragmented across the enterprise, leading to data that is redundant, inconsistent, and expensive to manage. Directories are being viewed by an increasing number of Oracle and third-party products as the best mechanism to make enterprise information available to multiple different systems within an enterprise. Directories also make it possible for organizations to access or share certain types of information over the Internet, for example, through a virtual private network. The trend towards directories has been accelerated by the recent growth of the Lightweight Directory Access Protocol (LDAP).

A specific type of enterprise information which is commonly proposed for storage in a directory is privilege and access control information. Both user privileges, represented as roles, and object constraints, represented as Access Control Lists (ACLs) listing those users who may access an object, may be stored in a directory.

Directory information which specifies users' privileges or access attributes is sensitive, since unauthorized modification of this information can result in unauthorized granting or denial of privileges or access to users. A directory which maintains this information on behalf of the enterprise must ensure that only authorized system security administrators can modify privilege or access information maintained in the directory. Oracle Internet Directory supports attribute-level access control and optional strong user authentication through SSL, and can be configured so that only specific users who are strongly authenticated are allowed to update directory information about user privileges or access.

Oracle8*i* introduced enterprise roles: centrally-administered privilege sets, maintained in Oracle Internet Directory, or in directories from selected partners which meet Oracle's security criteria. Enterprise roles enable strong, centralized authorization of users. Also, an administrator can add capabilities to enterprise roles (granted to multiple users) without having to update the authorizations of each user independently. Oracle Enterprise Security Manager provides one tool to centrally manage user definitions and assign roles, resulting in a lower cost of user administration throughout the enterprise. Another benefit of single station administration is that if security is easy to administer, organizations are more likely to implement strong security throughout the enterprise.

Schema-Independent Users

The schema-independent user, introduced in Oracle8*i*, extends the benefits of directory integration by allowing the database to delegate administration of user identity, as well as privilege, to the directory. A schema-independent user is a database user whose identity is maintained in a central LDAP repository; specifically, Oracle Internet Directory. When a schema-independent user connects to the database, the database queries the directory to determine if the user is registered there, and if so, to what database schema the user should be mapped, and what roles the user should obtain.

Schema-independent users reduce the administrative burden associated with managing users in the enterprise.

Suppose, for example, that there are 500 users of an application, who require access to data on several database servers in the enterprise. Instead of maintaining 500 different user accounts on each database, Oracle9*i* allows the system administrator to create a single shared schema (such as HRAPPUSER for the HR application), with appropriate privileges, on each database, and then create 500 enterprise users in an Oracle Internet Directory. When they connect to any specific database, these users are mapped to the appropriate schema on the database (e.g. HRAPPUSER), and inherit the privileges associated with the schema, as well as any additional privileges that are associated with the roles granted to them in the directory. Although these users share a common schema, individual schema-independent users 'identities are associated with their sessions by the database, and are used for access control or auditing purposes. Once created, these user accounts in LDAP can be used within multiple applications, as well.

The schema-independent user feature has a number of benefits. It reduces the administrative burden associated with managing users in an enterprise, and allows effective management of much larger communities of users than was previously possible. Moreover, it can provide a mechanism for integrating user account and privilege management across tiers in a multi-tier system, as long as the middle tier also supports management of user identities and privileges in the directory. In such a system, new users and their privileges can be registered once in a directory, and this gives them appropriate access to the middle tier as well as any databases in the enterprise that they need to access. In the future, it should be possible to build three-tier systems (e.g., web storefronts) in which new users can register themselves with a web server, and the web server then creates an entry for these users in the directory, giving them access to information in appropriate databases which pertain to them.

Password-Authenticated Enterprise Users

In Oracle8*i*, Enterprise User Security relies on client-side wallets to authenticate enterprise users. This requires SSL to establish secure channels between (i) the client and the server, and (ii) the database server and an LDAP-compliant directory. The authentication mechanism uses SSL and X.509 v3 certificates, requiring installation of Oracle wallets on both the client and the server.

Although this is a highly effective mechanism to ensure the integrity of the user authentication process, it requires SSL configuration and client-side wallets. Because this requires an X.509 certificate issued by a trusted Certificate Authority for each enterprise user, overhead can be significant for large organizations. Both SSL and an Oracle wallet must be installed on both the client and the server. This is a backwards-compatibility issue for certain earlier releases, and adds complexity to the setup and configuration process. Oracle9*i* adds new functionality that addresses these issues, while materially improving processing efficiency and ease-of-use.

In Oracle9*i*, Oracle Advanced Security implements password-based authentication for enterprise users, while eliminating the requirement for client-side wallets and most Secure Socket Layer (SSL) processing. (SSL is still required to secure connections between the database and Oracle Internet Directory.) With its reduced processing overhead, improved ease-of-use, and simplified setup and administration, this release is particularly useful for large user communities accessing multiple applications. Oracle Advanced Security supports enterprise user logins with password-based authentication for *all prior Oracle client versions.* Furthermore, enterprise users can use a single enterprise username and password to connect to multiple databases, if desired.

Password-authenticated enterprise users can enjoy single sign-on for Oracle client, including previous Oracle client versions.

ORACLE9I SECURE HOSTING AND SECURE EXCHANGES

Oracle’s Virtual Private Database feature, and the Oracle Label Security technology derived from it, provide very useful mechanisms for hosting and exchanges. Virtual Private Database provides a set of tools for enforcing fine- grained access control within the database. It can be configured to keep data from different organizations separate within a single database instance, so that organizations can share database tables but only see data which pertains to them. This makes it ideal for hosting, since a system administrator for a hosting company can set up and configure a single version of each application for which they provide hosted services, but use Virtual Private Database on the underlying application tables to provide separate virtual applications instances for each hosted customer. This can substantially reduce the costs associated with hosting. Because hardware, database, and applications instances can be shared, the costs associated with hardware, as well as installation and configuration of software, are lower than if physically separate instances were required for each hosted customer.

Oracle Label Security is particularly useful for hosting environments in which access to information can be formalized by means of sensitivity levels, access categories, or user groups. For these environments, Oracle Label Security makes it easy for hosting companies to define and administer label-based security policies. Oracle Label Security provides particular advantages for exchanges, because the label-based access policies include automatic, easy-to-administer "group" access embedded within a data label that can support communities of interest.

With Oracle Label Security, data labels remain with the data, making it easy to secure and enforce user privacy preferences across multiple eBusiness applications.

Many consumers are reluctant to purchase goods and services over the Internet because of privacy concerns. The label-based access policies of Oracle Label Security are also ideal for enforcing privacy concerns of users accessing eBusiness applications. Data can be labeled with an "opt out" provision for users who do not wish their data to be used for targeted marketing campaigns, or who do not wish their purchasing data to be sold. Data labels — and therefore users' privacy policies — remain with the data, making it easy to secure and enforce user privacy preferences across multiple applications.

SUMMARY

eBusiness depends on providing customers, partners, and employees with access to information, in a way that is controlled and secure. By providing deep data protection, Internet-scale security, and security mechanisms specifically targeted for hosting applications and exchanges, Oracle9*i* is an ideal platform on which to build and deploy eBusiness applications.

ORACLE®

Oracle9*i* Database Security for eBusiness
June 2001

Author: Mary Ann Davidson
Contributing Authors: John Heimann, Paul Needham and Kristy Browder

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.