3 Protecting Data in a Network Environment

docs.oracle.com/cd/B12037_01/network.101/b10777/protnet.htm

Introduction to Data Protection in a Network Environment

Security issues become more complex in a network environment. You must ensure that access to the network is controlled, and that data is not vulnerable to attack during transmission across the network. Many technologies are available to encrypt data and thus help to ensure its privacy and integrity. They ensure that:

- Data remains confidential.
- Data cannot be modified.
- Data cannot be replayed.
- Lost packets can be detected.

When multitier systems are involved, network access becomes even more complex. Users may access the network from a middle tier, in which case only the middle tier may be known to the database: the individual user's authorizations may be lost. To ensure confidentiality, the database must be able to identify the actual user who is accessing it from a middle tier.

Protecting Data During Transmission

The following sections describe the technology available to ensure data privacy and integrity during transmission:

Controlling Access Within the Network

This section describes different ways to control access within the network.

Middle-Tier Connection Management

You can configure a middle tier that manages the connections of very large user populations. To support a large number of users, you can configure multiple instances of Oracle Connection Manager. This product multiplexes multiple client network sessions through a single network connection to the database, increasing the total number of connections.

It is also possible to filter on source, destination, and host name. Thus you can ensure that connections only come from a physically secure terminal, or from an application Web server with a known IP address. (IP address alone is not enough for authentication, since it can be faked.) In this way you could allow connections from IP address `foo`, connecting to host `bar` for `payroll`.

Native Network Capabilities (Valid Node Checking)

In the case of a sensitive database, you may want to ensure that connections only come from certain points in the network. For example, a company might have a security policy saying that user `jausten` can access the payroll database, but only when she is present at work.

Database Enforced Network Access

You can also use Virtual Private Database (or secure application role) to limit access to the database from particular network nodes. Note that you would not want to make IP address a primary way of authenticating or authorizing users, since IP addresses can be faked. However, you can use IP address as an additional means of limiting data access for otherwise authorized users. For example, user Jane may have access to the `emp` table, but company policy may dictate that she is not allowed to access employee data unless she is inside the corporate intranet--perhaps even from a particular subnet for the HR department.

Encrypting Data for Network Transmission

Sensitive information that travels over an intranet or the Internet can be protected by encryption. Encryption is the mutation of information into a form readable only with a decryption key. Encryption is a powerful security mechanism because it can make decryption mathematically infeasible if you do not possess the decryption key.

Consider, for example, an Internet buyer who wishes to purchase a company's product by using a credit card in a secure fashion. The buyer's credit card number is encrypted with an encryption key. The encrypted credit card number is sent across the network to the database. Encryption scrambles the message, rendering it unreadable to anyone but the recipient. The server decrypts the message with a decryption key and reads the credit card number.

Note that the secrecy of encrypted data depends on the existence of a secret key shared between the communicating parties. Providing and maintaining such secret keys is known as key management. In a multiuser environment, secure key distribution may be difficult; public key cryptography was invented to solve this problem.

Encryption must address all communications with the database, including transmissions from clients and transmissions from middle tiers. It must also secure all protocols into the database.

Encryption Algorithms

Table 3-1 lists encryption algorithms that have become industry standard for the encryption and decryption of data.

Table 3-1 Encryption Algorithms

Algorithm	Characteristics
RSA Data Security RC4	Allows high-speed encryption for data privacy. By using a secret, randomly generated key unique to each session, all network traffic is fully safeguarded--including all data values, SQL statements, and stored procedure calls and results. The client, server, or both, can request or require the use of the encryption module to guarantee that data is protected.
Data Encryption Standard (DES)	Uses symmetric key cryptography to safeguard network communications. DES is required for financial institutions and many other institutions.
Triple DES (3DES)	Encrypts message data with three passes of the DES algorithm. 3DES provides a high degree of message security. However, it entails a performance penalty, the magnitude of which is dependent upon on the speed of the processor performing the encryption. 3DES typically takes three times as long to encrypt a data block as compared with the standard DES algorithm.

Data Integrity Checking

In addition to encryption, there are integrity algorithms that can ensure that data has not been tampered with or packets replayed. A database can use these algorithms to detect corruption in data blocks. Table 3-2 lists industry standard integrity algorithms.

Table 3-2 Integrity Algorithms

Algorithm	Characteristics
MD5 Checksum	Provides data integrity through hashing and sequencing to assure that data is not altered or stolen as it is transmitted over a network
Secure Hash Algorithm (SHA)	Similar to MD5, but produces a larger message digest, for greater security

Secure Sockets Layer (SSL) Protocol

The Secure Sockets Layer (SSL) protocol, developed by Netscape Corporation, is an industry-accepted standard for network transport layer security. SSL is supported by all currently available Web servers and Web browsers. It is also gaining acceptance for other protocols, including LDAP and IMAP. The SSL protocol provides authentication, data encryption, and data integrity, in a public key infrastructure (PKI).

SSL addresses the problem of protecting user data exchanged between tiers in a three-tier system. By providing strong, standards-based encryption and integrity algorithms, SSL provides system developers and users with confidence that data will not be compromised in the Internet. 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 an application Web server before they provide the server with sensitive information, such as credit card numbers.

Firewalls

To eliminate potential weak points in the network infrastructure, you may opt to pass data from protocol to protocol without the complexity of decryption and re-encryption. To do so securely, you must have some way to securely transfer data across network protocol boundaries.

The Internet enables you to connect your corporate intranet to a broad public network. Although this capability provides enormous business advantages, it also entails risk to your data and your computer system. One way of protecting the privacy and integrity of your system is to place a firewall between the public network and your intranet.

A firewall is a single point of control on a network, used to prevent unauthorized clients from reaching the server. It acts as a filter, screening out unauthorized network users from using the intranet. It does this by enforcing access controls based on the contents of the packets of data being transmitted, and can thus protect against attacks on individual protocols or applications. Firewalls are rule-based. They have a list of rules that define which clients can connect, and which cannot. They can compare the client's host name or IP name with the rules, and either grant the client access, or not.

Ensuring Security in Three-Tier Systems

The following sections discuss security issues in multitier systems:

Proxy Authentication to Ensure Three-Tier Security

An important security feature for three-tier systems is the ability to proxy authenticated user identity from a middle tier to the database. This feature (also known as *n*-tier authentication) enables you to identify the real user who is accessing the database through a middle tier. It authenticates users and machines by way of a database password or other credential, without the overhead of a separate database connection. It protects data on the server by ensuring that unauthorized users cannot access data on the server over the Internet or through a middle tier. It ensures accountability by keeping track of what users have logged on to an application through an intermediate tier, so that it is possible to trace who has done what in an application. Scalability is further improved through the introduction of support for enterprise users.

Java Database Connectivity (JDBC)

You can use Java to transmit data securely in a three-tier environment. Java is the language of the Internet, and also the language of OLAP applications. Application developers use Java to build applications and applets. As an object-oriented, platform-independent, network-based, and secure language, Java is fast superseding C++ and Visual Basic as the language of choice for application developers.

JDBC (Java Database Connectivity) is an industry-standard API (Applications Programming Interface) that allows Java programs to send SQL statements to an object-relational database such as Oracle. JDBC enables a middle tier server to access a database on behalf of a client user by establishing a lightweight session for the user.

Java applets can thus transmit data over secure channels. You can have secure connections from middle tier servers with Java Server Pages (JSPs) to the database. This enhances security because:

- Every protocol can be secured.
- JDBC-Oracle Call Interface and thin clients can be supported.
- Two-tier and three-tier architectures can be supported.

There are two ways to implement Java security and negotiate algorithms:

- Hard code it into your JDBC client application
- Configure it like native network cryptography

JDBC-Oracle Call Interface Driver

The JDBC-Oracle Call Interface (JDBC-OCI) driver can be used for client side use with an Oracle client installation.

JDBC Thin Driver

The JDBC Thin driver is a Type 4 (100% pure Java) driver that uses Java sockets to connect directly to a database server. It has a lightweight Java implementation of Oracle Net called Java Net.

The Thin driver does not require Oracle software on the client side. It does need a TCP/IP listener on the server side. Use this driver in regular Oracle Net listener Java applets that are downloaded into a Web browser. The Thin driver is self-contained, but it opens a Java socket, and thus can only run in a browser that supports sockets.