Securing Database Servers

Database security for enterprise information systems and security professionals

Introduction:

Database servers are the foundation of virtually every Electronic Business, Financial, and Enterprise Resource Planning (ERP) system, and frequently include sensitive information from business partners and customers. In spite of the importance of preserving data integrity and security on these systems, databases typically have not been subjected to the same level of security scrutiny as operating systems and networks. Data integrity and improper access can be compromised by many factors, including complexity, insecure password usage, misconfigurations, and unrecognized system backdoors, making imperative regular use of an adaptive database security solution.

Why is Database Security Important?

To protect access to an organization's sensitive information and digital assets.
A large majority of any organization's electronic digital assets are stored in off-the-shelf relational database products. Businesses and government organizations use these database servers for personnel information such as employee payroll and medical records for which they have responsibility for privacy and confidentiality. Database servers hold sensitive financial data, past and future, including trading records, business transactions, and accounting data. Strategic or classified information such as proprietary technical and engineering data - even marketing plans - must be guarded from competitors and unauthorized internal access. Database servers also include detailed customer information including financial accounts, credit card numbers, and the trusted data of business partners.

Databases are extremely complex systems and difficult to correctly configure and secure.
Database server applications can be amazingly complex to master – on a level that rivals that of the operating systems they run on. Database systems such as Oracle, Sybase, and Microsoft SQL Server include the following features: user accounts and passwords, auditing systems, privilege model and specific permissions for control of database objects, built-in commands (stored procedures/packages), unique scripting and programming languages (usually vendor specific derivatives of SQL), middleware, network protocols, patches and service packs, and powerful database management utilities and development tools. Most DBAs have a full time job administering the complexities of these systems. As a result serious security vulnerabilities and misconfigurations frequently go unchecked or completely undetected. Therefore, just as the traditional security community has mostly ignored the topic of database security, database professionals usually don't consider security as one of their primary responsibilities. The philosophy of "Adaptive Network Security" **–** viewing security as an ongoing "process" instead of a onetime checklist **–** has not yet been embraced by many database administrators.

Securing the network and operating system on a database server is critically important, but insufficient to protect that database server.
This is a common misconception among experienced security professionals, who assume that, once they have assessed and locked down critical network services and operating system vulnerabilities, all applications on those servers will also be secured. Modern database systems have a wide array of features and capabilities that can be misused or exploited to compromise the confidentiality, availability, and integrity of data. To begin with, all modern relational database systems are "port addressable," which means that anyone with readily available query tools can attempt to connect directly to the database, bypassing security mechanisms used by the operating system. For example, Oracle 7.3 and 8 can be accessed via TCP/IP on their default ports of 1521 and 1526. Most database systems also have well-known default accounts and passwords, which provide varying levels of access to database resources. These two simple data points combined could probably compromise an embarrassingly large number of critical database systems. Unfortunately, the opportunities to compromise database security does not end there for knowledgeable intruders.

Poor Database security not only compromises the database, but the server operating system and other trusted systems, too.
This is another not-so-obvious reason why database security is important – a database system may itself provide the mechanism to compromise an entire network's security infrastructure. As an example, a company may have a database server that keeps an inventory of all technical manuals, documentation, and white papers. The information in the database is not considered mission critical, so its security is not a priority. Even running on a well-secured operating system, an intruder could use access to the database to gain access to the local operating system through powerful built-in database features such as "extended stored procedures." These procedures can give Administrator-level command line access to the underlying operating system and full access to all of its resources. If this particular database system has trust relationships with other servers, the intruder could potentially compromise the security of the entire network domain.

Databases are foundation of new Electronic-Business, Enterprise Resource Planning (ERP) and other critical business systems.
While much of the focus of E-Commerce and E-Business is on Web servers, Java, and other emerging technologies, it is important to remember that the vast majority of these consumer-oriented and business-to-business systems are based on relational databases behind the Web servers. Security in these systems is directly related to systems availability, data and transaction integrity, and confidentiality. Downtime and lack of system availability can be damaging not only to a business, but also to a company's reputation. By necessity, these systems are more open to the dangers of intruders, yet the responsibility for confidentiality of business partners and customers' sensitive information has never been greater. In addition, ERP and management systems such as SAP R/3 and PeopleSoft are built on these same standard database systems. Unmanaged security vulnerabilities have a direct correlation with costly downtime, system integrity issues, and customer confidence.

What types of security vulnerabilities do I need to look for?

Traditional database security focuses only on user accounts, roles, and operating permissions on specific database objects, such as access to tables and stored procedures. A thorough security analysis of a database system must be much broader, assessing potential vulnerabilities in all possible areas, as broadly outlined in the categories below.

Risks associated with vendor-supplied software – bugs, missing operating system patches, vulnerable services and insecure choices for default implementations and configurations.

Risks associated with administration – security options available but not used correctly, risky default settings, improper granting of excessive privileges to users or unauthorized changes to the system configuration.

Risks associated with user activity – insufficient password strength, inappropriate access to critical data and malicious activities such as stealing contents of databases.

These categories of risk can apply to network services, operating systems, or the actual databases themselves. All of these elements need to be considered when securing database servers.

Database Security – Vulnerability Areas & Examples

Listed below are a few examples of the wide variety of database server vulnerabilities and misconfigurations that can exist in critical database servers.

Lack of security feature maturity – Most popular relational database systems have been in existence for well over ten years and are mature products with powerful feature sets and scalability. Unfortunately many features that IT and security professionals require and take for granted in operating systems and networks are simply not available in most of the popular database systems in use today.

No built-in database standard security features to address:

	MS SQL Server	*Sybase*	*Oracle 7*	*Oracle 8*
Account Lockout Facility	***no***	***no***	***no***	***yes***
Rename Admin Account	***no***	***no***	***no***	***no***
Require Strong Passwords	***no***	***no***	***no***	***yes***
Stale Accounts	***no***	***no***	***no***	***no***
Password Expiration	***no***	***yes***	***no***	***yes***
Login Hours Restrictions	***no***	***no***	***no***	***no***

For example, the table above lists features that most IT professionals would expect or require in an operating system, but are not present in a database server's standard security offerings. Since these databases are port addressable, the core operating system security mechanisms are not applied to direct network connections to the database. Some products, such as Microsoft SQL Server, allow the more powerful

Windows NT security mechanisms to address some of the shortcomings listed above. However, the majority also run MS SQL Server standard security for backward compatibility in environments that are not 100% Windows NT. Implementation is another concern. If an organization is implementing Oracle 8, how does the administrator know if security features are actually being used? Are they being consistently implemented across the organization?

The combination of several of these features makes the implications even more serious. Since System Administrator accounts ("sa" for SQL Server and Sybase, and "system" and "sys" for Oracle) can't be renamed, and if there is no password lockout available or configured, intruders can launch brute force dictionary login attacks against the database server with nothing to prevent them from patiently and persistantly trying to break into the server at the highest level access.

Database Password Management – In the standard security provided by most database systems, there is no mechanism to ensure that individual users are choosing strong – or any – passwords. This fundamental security issue requires careful monitoring. There is also the additional issue of managing and securing a whole list of additional system passwords. For example, Oracle database systems have over ten specific default user accounts and passwords, and additional unique passwords for managing critical database operations such as booting the Oracle database, access to the network listener process and remote database access privileges. If compromised, many of these system passwords would give an intruder full access to the database system, yet they are stored in ordinary text files on the operating system. A few examples:

Oracle Internal Password – the Oracle Internal password is stored in clear text within the following file "strXXX.cmd", where XXX is the Oracle System ID or SID, which defaults to "ORCL". The Oracle Internal Password is used for Oracle database start-up processes, and gives complete access to database resources. This file needs be properly secured for Windows NT based Oracle implementations.

Oracle Listener Process password – Password used to start and stop Oracle listener process, which routes all incoming traffic to appropriate Oracle instances on a system. A strong password needs to be chosen for this value to replace the system default, and the permissions must be secured on the "listener.ora" file, where the password is stored for all Oracle implementations. Improper access to this password could allow an intruder to create a Denial of Service attack on an Oracle based electronic business site.

Oracle Internal Password – ***"orapw" File Permission Control*** – Oracle Internal password and passwords of accounts granted the SYSDBA role are stored in the "orapw" text file. Privileges to this file should be restricted on Unix and Windows NT implementations of Oracle even though it is encrypted. If accessed, the encrypted file is potentially vulnerable to brute force attacks.

These are just a few examples of how critical administrator and system passwords and accounts can be compromised in unexpected ways. Please note that password management security issues are by no means unique to the Oracle database environment, and apply to virtually all major database vendors

Operating system back doors - Most database systems have powerful features that, while convenient for DBAs, are also potential backdoors into a database server's host operating system.

As mentioned previously, an intruder who has compromised a Sybase or SQL Server "sa" password can potentially gain system rights to the underlying operating system by using "extended stored procedures". By logging in as 'sa', an intruder has use of the extended stored procedure **xp_cmdshell**, which allows a Sybase or SQL Server user to run an operating system command as if that person was running a command prompt at the server console. As an example, the following SQL commands can be used to add a Windows NT account called 'hacker1' with a password of 'nopassword', and then add 'hacker1' to the 'Administrators' group:

```
xp_cmdshell 'net user hacker1 nopassword /ADD'
go

xp_cmdshell 'net localgroup /ADD Administrators hacker1'
go
```

Now the unscrupulous intruder is a Windows NT Administrator (let's hope this SQL Server is not on a domain controller). This simple attack works because the commands are submitted to the operating system using the Windows NT account under which the MSSQLServer service runs under. By default, this is the 'LocalSystem' account, which is the most powerful account on the local Windows NT system. Another way a hacker might use SQL Server to compromise the operating system's security is by using the stored procedure **xp_regread** to read the encrypted Windows NT SAM password database out of the registry. There are several free Windows NT password crackers that make it important to keep these encrypted Windows NT passwords secure. Below is an example of how an intruder can get at this information:

```
xp_regread 'HKEY_LOCAL_MACHINE','SECURITY\SAM\Domains\Account', 'F'
```

Note that reading the encrypted passwords out of the registry is something even the local Windows NT "Administrator" account cannot do. SQL Server allows this data to be read from the registry because SQL Server by default runs using the 'LocalSystem' account.

Oracle database systems also include powerful features that can be used to gain direct access to the operating system's native file system. For example with proper access, the **UTL_FILE** package allows users to read and write files to the host operating system. The **UTL_FILE_DIR** profile variable can be misconfigured or intentionally set to allow Oracle users to use the UTL_FILE package to write anywhere in the file system, which also has the potential to compromise the host operating system.

Auditing – The depth of information and events that can be recorded by the auditing systems of relational database systems varies from basic to very granular and detailed, but auditing systems only provide valuable security and alerting information if properly used and configured. These features can provide early warning signs that intruders are compromising the security of specific database servers, and provide valuable clues for detecting and repairing any damage.

Trojan Horses – Trojan horses in operating systems have been around for years, but Database Administrators need to be aware of the threat they pose to system stored procedures. One well-known Trojan Horse modifies the password changing stored procedure, notifying the intruder of new passwords as they are updated. For example, an individual can add several lines to the sp_password system stored procedure to log new passwords to a table, send the password in an e-mail, or write the password to an external file for later review. This procedure continually grabs passwords until the intruder catches the "sa" password being changed – leading to the accomplishment of further break-ins going undetected. An intruder or disgruntled employee only needs to gain access to a system once to place this trojan horse and steal a stream of future passwords.

Conclusion

Security professionals, auditors, DBAs, and E-business architects need to be aware of database security as they deploy critical business systems. To understand your organization's security posture and potential exposure, a thorough assessment of the database servers and their host operating system should be performed before system deployment, and then reprocessed on a regularly scheduled basis. The application of information risk management principles to monitoring, detecting, and responding to security vulnerabilities should be required for all database systems

While this white paper focuses primarily on database specific vulnerabilities, ISS recommends familiarity with network and host based security assessment tools for evaluating network services, operating systems, and network infrastructure devices, such as routers and firewalls. A good place to start is the ISS white paper entitled "**Network and Host-based Vulnerability Assessment"** available at:

http://www.iss.net./prod/whitepapers/nva.pdf

More detail on E-Business issues can be found in **"Secure E-Business"** available at:

http://www.iss.net./prod/whitepapers/securityebus.pdf

About ISS

ISS is the world's leading provider of security management solutions for the Internet. ISS protects digital assets and ensures the availability, confidentiality and integrity of computer systems and information critical to e-business success. Combining its best-of-breed SAFEsuite security software, comprehensive ePatrol monitoring services and industry-leading X-Force security research, ISS provides customers with complete security management solutions. Serving as a trusted security provider, ISS protects more than 5,000 customers including 21 of the 25 largest U.S. commercial banks, 9 of the 10 largest telecommunications companies and over 35 government agencies. Founded in 1994, ISS is headquartered in Atlanta, GA with additional offices throughout North America and international operations in Asia, Australia, Europe and Latin America. For more information, visit the ISS Web site at www.iss.net or call 800-776-2362.