Global Information Assurance Certification Paper

Database security in high risk environments
Joaquin A. Trinanes. Version 2.0
September, 2001-09-14

During all ages, pirates hunted for treasures. Violence was usually involved to acquire them. In today's economy, corporation databases symbolize one of the most valuable assets that the modern bandits try to break in. Methods have also changed and physical presence is frequently not required. A simple computer attached to the Internet can be the only weapon needed to gain access to a "treasure". In this work, we will describe some of the methods in use to protect databases, desirable techniques to improve data confidentiality and integrity, and new viewpoints to consider in the beginning of the e-commerce era.

Data, to be or not to be (secured)?

It is evident that data security principles must be applied to sensitive information. Even more, databases, independently from their value, must share some basic security standards to protect information integrity. Government, research, corporations and other organizations keep large volumes of data which are not expected to be available to non-authorised users. And if someone can access them, data should be unreadable and absolutely incomprehensive.

Data accessibility is a major goal (and concern) in database security. Many organizations can not work properly if databases are down, they are what we know as mission-critical systems. To put the data available implies to provide the security mechanisms to ensure authentication, authorization and auditing procedures. Authentication means that user identity must be truly verified, commonly through a password only known to the user. This is a critical phase, the foundation of the security strategy. After this first step has been completed, the system must determine the resources that the particular user id can access to. This is the authorization phase and all the tasks involved are often referred as user security administration. Finally, to detect possible intruders and ensure data integrity, auditing utilities must be activated.

While on routing from server to receiver, data passes through different devices where, if security policies have been not applied or are defective, a third-party can get access to the packets. This is potentially dangerous with some kind of information, such as, e.g., credit card numbers, payrolls, social security numbers, and medical records, to name a few ones. Obviously, this is a security threat that must be taken into strong consideration We must ensure that data can be only seen by the same individual we sent the data to, while avoiding data corruption by a third party. To achieve the former, encryption must be applied. Data integrity is usually ensured by means of certificates and public key encryption. Another important aspect is named non-repudiation. This is extremely important in e-business and it avoids that a sender can deny to have sent the information.

Internet is a public network and risks can be anywhere data crosses, exposing passwords, accounts, personal information to unauthorized individuals. Any security policy must begin by analysing and managing the risks involved. The analysis phase implies to identify and value the assets, and to assess the threats, vulnerabilities, defences and risks. It provides the basis for risk evaluation, treatment and acceptance. In

the management phase, there are three possible approaches commonly accepted. The first involve to reduce the risk by reducing the vulnerabilities and/or threats and/or the impact on the system and/or, improve recovering from a potential attack. The second performs a risk transfer to another party by, e.g. insuring the assets . Finally, risk can be accepted and no countermeasures are applied, usually because of the high cost associated with them (Are you going to spend $100 to potentially secure $100-$200-$300-… valuables?). Another tactic is to eliminate the risks, but this is unpractical as it requires to renounce to some of the primary objectives ("I need this online; it has risks; I do not need this online.").

Previous analysis allows to create a picture of the whole system risks, allowing to take decisions related with the amount of security to implement (we could also refer it as amount of money to spend). In a risky environment, an accurate analysis is extremely important. All security aspects must be properly enforced, specially those related with database issues, usually placed in a secondary row when compared to operating system and network security investments. It is often wrongly thought that securing both of them will prevent any kind of attack addressed to the database server. Most of the relational databases are "port addressable" and any user can try to access to them by connecting directly to the port, usually a default port. Joining this to the fact that some of these databases have some well-known default accounts which can be the targets of a brute-force dictionary attack, it all together creates a potential very dangerous situation. Databases usually provide authentication, authorization and auditing on a basic level. So, most of the databases do not ensure password robustness and they are even stored as plain ASCII text. Some of these passwords can compromise the whole system allowing in some cases to access the underlying operating system with system privileges. Database administrators usually perform both database operation and administration duties and it is not atypical that a person who is a first-class SQL expert, performs really bad as a system manager, and vice versa. A good practice is to separate tasks by roles, and by applicable user community, which is commonly known as segregation of duties. It also prevents that a single individual could undermine a critical process. Another essential requirement is the "least privilege" principle, where every user will receive the lowest level of permission needed to perform their habitual tasks. It limits database damage from improper or accidental activity. Latest releases of some of the most popular relational databases (Oracle, DB2), provide even row level security and users can access rows of data in tables they are authorized to. Certainly, proper account management is important. Attention must be given on applying appropriate password settings, implement account lockouts after a number of failed login attempts (to prevent an unlimited number of attempts to guess a password), changing default passwords, setting access level to data, and managing account life cycles. Former employees must have their account disabled and users must change their passwords periodically. If not, the chance of they being compromised increases with time. When a new password is activated it must observe some basic security rules to prevent weak passwords (minimum length, alpha-numeric characters, dictionary check, …). Multiple passwords should be needed to access to the server and the information in it. This is the principle of compound security and will strengthen defences against unauthorized users.

In many databases, auditing procedures are often inactivated to avoid decreasing the performance of the system and then, there are no records to study any illegitimate activity in the database. This does not mean that recording everything is the optimum

solution. Depending on the system nature, an intelligent auditing strategy can be built to highlight security problems while maintaining database performance.

Encryption is commonly associated to information moving from one side to another. On a high risk database environment, encryption must be also enabled to the lowest level, that is, to the stored data. It adds an important layer of protection that enforces security. Any user trying to access the data not only need the right password, but the encryption key as well. One advantage of this schema is that files can be unreadable to people that have access to the database, such as a system administrator, but no databases privileges. Database encryption affects performance and a compromise solution must be found between performance and security, only encrypting tables, or columns with sensitive information.

There is a need to protect databases from accidental data loss. A general backup and recovery strategy must be designed depending on various factors, such as database size, volume of changes, and resources available. Attention must be paid when choosing the backup type (incremental, full) and testing the whole set of procedures to recover the system in case of disaster, and in a timely manner. Backup utilities, usually integrated into the database package (e.g., On-Tape and On-Archive in Informix, RMAN in Oracle), guarantee that backup procedures can be carried out while maintaining database security and referential integrity. But database sizes are growing to an exponential rate and the backup tools lack of performance and automation is a problem. Multitasking, backup and restore parallel processes are not generally supported, tapes must be changed manually, maintenance is problematic, and operator must specify the data to backup or restore. These drawbacks can be put aside by using specialized software, such as Legato Networker. Parallelism is strongly suggested, mainly for critical applications, as it ensures minimum service disruption. The latter can be also improved by using high speed backup devices.

It is thought that most of the attacks come from outside but, in the real world, a large percentage of the security breaking incidents are made by insiders, people working in the same organization with extra knowledge about the database structure and security policy. If the previous database security recommendations are followed, damage caused by insider's activities can be limited and audited. Right security policies largely decrease risks by reducing vulnerabilities and strengthening defences and countermeasures. A database server with important information is tempting for many people and appropriate measures have to be taken. They include the physical security of the server and the number of people accessing it.

Moving to the Internet

Databases are the basis of e-business, e-commerce, Enterprise Resource Planning (ERP) and other sensitive activities. Nowadays, with the development of electronic commerce on the Internet, among others, security is a must. Money amounts involved in this business are huge and estimations are than in 2002, e-business will move close to $1 trillion. This kind of market attracts data, many of them very sensitive, that need to be secured. Philosophically, this means that data domain is not the Intranet alone anymore, but also the Extranet, where anyone may perhaps access them. There are also new legislation and guidelines that regulates data privacy, such the EU 95/46/EC Directive on Data Privacy (commonly known as Safe Harbor), GLBA (Gramm-Leach-Biley Act),

HIPAA (Healthcare Insurance Portability and Accountability Act), CISP (Cardholder Information Security Program) and the BITS Voluntary Guidelines for Aggregation Services.

In a serious online application, a typical configuration locates the database with the sensitive information behind a firewall. It will be accessed from an application-server also located behind a second firewall, which will receive the Web server requests. This three-tier design isolates the Web-server from the database, isolating the database server from the outside users by two dedicated private networks. Only the Web server can communicate through the firewall with the application-server, and only this can communicate with the database. This configuration is relatively secure and special attention must be paid on securing the information sent to the client from the Web server, the Web-server itself, and the database/application-server system. The application-server will incorporate the event logging and the security analyser that recognizes unauthorized attempts to log into an account.

Restricting our attention to the Web server-client interaction, the Secure Sockets Layer protocol (SSL), whose last release was named TLS (Transport Layer Security), a non-proprietary protocol developed by Netscape, which lies between the application (where the HTTP, FTP, SMTP protocols reside) and the TCP/IP layers, at the top of the transport layer, is a de facto Internet standard for authentication and encryption. It is very easy to identify a SSL-based Web page as generally they have an URL beginning with https, instead of http (this does not apply necessarily to framed pages). Most of the web browsers will inform the client when passing from a secure to a non-secure environment, and back, and it is the user's responsibility to avoid non-secure transactions where personal and other important information is transferred. SSL provides server and client (optional) authentication. The first feature allows a client to validate the server's certificate and public ID issued by a trusted CA. The second one, is virtually the same as before but with server and client names swapped. Both of them, can now identify the other part as a trusted host and send through the network the data, such as account numbers, payments and other important information. To prevent data from being altered when moving from one side to another and ensure confidentiality, encryption must be used. SSL uses an encryption algorithm, usually RC4, IDEA or Triple-DES, to cipher the data exchanged between the server and the client. The RC4 or IDEA session key is also ciphered using a public-key algorithm, most of the times RSA. The session key is different for each transaction and so, if any of them is "hacked", they will be invalid on future transactions. MD5 and SHA are the hash algorithms used. If our application is going to be used in banking, e-commerce or other activities where security is critical, RC4-256 or Triple-DES-168 are formidable encryption algorithms. There are some restrictions for using these technologies in other countries different from USA and Canada, although financial institutions can overpass them. If this is not the case, RC4-40 and DES-56 are still relatively secure. In the near future, the AES algorithm will allegedly improve robustness in the encryption procedure.

SSL record and SSL handshake protocols are part of SSL. The former defines the format used to transmit the data. The data protocol section has three components: MAC-data (the message authentication code), actual-data (the data being sent) and padding-data (to complete the message when block cipher is used).

The SSL handshake protocol uses the record protocol to exchange messages between the server and the client and authenticate themselves, choose a cryptographic algorithm and establish an encrypted SSL connection. Briefly, the handshake phase follow these six steps:

1. Hello phase. Server and client choose the set of algorithms to maintain confidentiality and authentication.
2. Key Exchange. Both sides share a master key.
3. Create a session key. It will be used to cipher the messages.
4. Server authentication. The client authenticates the server (only when RSA is used).
5. Client authentication. Server ask for a X.5009 certificate to the client (if client authentication is needed).
6. Finished. The secured session can begin.

Other way to cipher and authenticate is using the Secure HyperText Transfer Protocol (SHTTP). It has been developed by EIT and it is an application layer protocol. This protocol proposes a new document extension (.shtml) and is a secure add-in to HTTP. Through a GET, a client requests a document, providing information about the cipher type it can manage and where the server can find its PK. If the user is authorized to access the document, the server ciphers it, and sends it to the client, which will use its secret key to view the document. Messages exchanged between server and client includes security options and algorithm information, such as certificate type, data symmetric data block cipher algorithm, symmetric header cipher algorithm, and key exchange algorithms. It is important to highlight that one of the encryption methods that are available with SHTTP is the popular PGP.

SHTTP services are integrated with HTTP, and can be used for HTTP connections only. Besides, not all browsers and servers support this protocol. As SSL is in a lower layer, it can be used by other protocols as well and it is transparent to users and applications. An important point to emphasize is these protocols are designed in a very different way and they can be utilized together, providing a robust environment for implementing e-commerce and other applications requiring elevated security on the Internet.

There are other protocols for Web transactions, like SET (Visa's and Mastercard's Secure Electronic Transaction). It has been designed to improve the payment process while ensuring that the purchaser is an authorized user and the vendor an authorizer acceptor. They are provided with digital certificates and when a transaction is made, each side are sure who they are and so, there is a higher level of security. A variation is named MOSET, where there is a combination of SET and SSL. The first protocol is used between the vendor and VISA/Mastercard/Bank and the second one between the client and the vendor. These solutions are not very popular yet, but it is expected that the number of users will increase in the future. They are not generic solutions as they are addressed almost exclusively to electronic payment environments.

Conclusion

The solutions shown above try to guarantee, in the greater degree as possible, both database and data security. The importance to do so has been also highlighted and there is a general recognition that these methods must be applied in the highest level, mostly

in databases with very sensitive information. Availability of funds to apply a good security policy is an important feature, but human factor is also essential. Good DBAs and security managers can handle system resources and improve security by applying some basic low-cost techniques, reducing vulnerabilities, eliminating backdoors, ... In an ideal situation all risks would be eliminated. But in the real world, risks can not be eliminated, they must be managed instead.

References

- BrainTree. "Client/Server Database Security". URL: http://www.bti.com/Whitepapers/3-Tier_Data_Security/WP_Download/bti_down2/Cswp.pdf
- ISS. "Network and Host-based Vulnerability Assessment". URL: http://www.iss.net/prod/whitepapers/nva.pdf
- ISS. "Secure E-business". URL: http://www.iss.com/prod/whitepapers/securityebus.pdf
- ISS. "Securing Database Servers". URL: http://documents.iss.net/whitepapers/securedbs.pdf
- Le Tocq, Chris & Young, Steve. "SET Comparative Performance Analysis". 2 Nov 1998. URL: http://www.setco.org/download/setco6.pdf
- Legato. "Manager's Guide to Informix Database Protection". URL: http://www.iqproducts.de/iqproducts/whitepapers/pdf/guide_informix.pdf
- Mattsson, Ulf. "Secure Data Technology Overview". 27 Apr 2001. URL: http://education.protegrity.com/downloads/SecureData_IBMv1.pdf
- Pentasafe Security Technologies, Inc. "Common Vulnerabilities in Database Security". May 2001. URL: http://www.pentasafe.com/whitepapers/pentaSafe_wp_common_vulnerabilities.pdf
- Sybase Inc. "What Backup, Recovery, and Disaster Recovery Mean to Your Adaptive Server Anywhere Databases". 15 Jun1999. URL: http://www.sybase.com/detail/1,3693,47877,00.html
- Turner, Tom. "Securing the Database: What are the Issues?". URL: http://www.itaudit.org/forum/security/f218se.htm