Abstract

The concept of enterprise information security is related to who is allowed to access information and what they can do (such as, read, write and execute privileges) with that information in various forms. Whatever the form the data is in, such as data in use, data in transit and data at rest, the restrictions on access should be considered before anything else.

Confidentiality means keeping the right person in, and wrong person out. Unfortunately, there is no fool-proof mechanism to control against an unauthorized access when it comes to internal users with powerful privileges (such as, a system administrator or a database administrator (DBA)).

Access to enterprise information must be planned and limited for a variety of users, whether external and internal, for a variety of reasons. Among these groups, internal users with powerful privileges are the most difficult to manage from an access point of view. Because of their jobs, these type of users will always have an easy access to any type of data in the system. Since there is no such thing as a foolproof access control for these type of users, in this paper we recommend the consideration of the data encryption as the second line of defense.

Unfortunately, the encryption of a database as a whole creates additional performance issues. To avoid the performance related issues, the possibility of encrypting selected components of a database, such as rows, columns or even cells should be considered to protect the data from unauthorized accesses.

INTRODUCTION

Information security issues have been a serious concern for computer scientists since the early days of computing. The classic triad of “CIA”, confidentiality, integrity and availability, is the most commonly used framework for this purpose.
Confidentiality, or privacy requires limiting access (or viewing) to authorized users. Most security experts relate the concept of confidentiality to disclosure, but confidentiality may be lost by observation, whether that observation is voluntary or involuntary.

Integrity is the capability of the system to maintain the original content intact (in other words no unauthorized modification of the data has taken place). Integrity means a state of wholeness, completeness, and soundness.

Availability is related to the usability of information for a purpose, or capability to reach the data/system at the moment we want it. For instance a distributed denial of service (DDoS) attack or changes in the name of a program file may stop the usage of the system for a long period of time.

At the hardware level, the "read", "write" and "execute" privileges given to various groups of data and system users summarize the whole process of information security. The read privilege determines the capability to reach the information or system. The write privilege determines the power to modify the existing data or the system. Modification of data includes changing or deleting a file or modifying an existing program. An unauthorized modification to the data or the system indicates the lack of integrity for the data or the system. Execute privilege is related to the authorization to run a system or program.

When we look at these privileges, it is clear that the read privilege supersedes any other privilege. If an unauthorized person or group of persons cannot reach the data or the system they cannot harm them. If we want to be sure of data/system security we must control the “read” privileges.

These types of issues have been a serious concern for computer scientists since the early days of computing. Many alternatives to a secure system was offered. However, very quickly they learned that no single method could solve all security problems mentioned above. Instead of offering a single protective mechanism, a multiple layer of defense was agreed upon. Just like a medieval castle with many different walls of defense (drawbridges, moats, stairs, inner and outer walls), computer systems need many layers of defense. The more layers we could put between the data and a malicious attack the more possibility we would stay safe and secure.

In the following sections we look into these layered defense approach.

I. THE FIRST LAYER: ACCESS CONTROLS

a) The Physical Access Controls (Hardware Level Defense)

The first line of defense is to protect the computing equipment and computer network from an unauthorized physical access. A restrictive access policy to a computer center, e.g. locked doors or a palm reader or a voice recognition system, could be used for this purpose. If and only if, the computer center is on an isolated island and separated from the rest of the world, it could be assumed safe. Since it is impossible to obtain that isolated island, we settle for the second best: keep the system away from a heavy traffic areas. Obviously, this does not

b) The Logical Access Controls: Authentication and Authorization

Today, to be able to attack a system close proximity is not needed. Most systems are exposed to the rest of the world through various types of computer networks, such as LAN or the Internet. In this new environment, the computer can be accessed from anywhere. This kind of access is called “logical access”.

The logical access control is obtained in a two step process, authentication and authorization.
Authentication provides answers to the question of "Who is the user?" or "Is the user really who he/she claims to be?" The verification process typically challenges the user to provide his/her unique information (his password, fingerprint, etc.). Computer verifies accuracy of the information provided with the information stored in the system. If the answer matches the stored data, the user is considered to be authenticated.

This verification may be based on one or more challenges. Password or passphrase verification is an example of one factor authentication. Demonstrating knowledge of password does not directly authenticate a user. All it does is authenticate knowledge of password. For instance, guessing a password can lead to impersonation of one user by another.

When a more secure environment is needed authentication may be based on possession of a token. In this two factor authentication format, the user has at least two challenges. After the user proves that he has the possession of a token, he has to prove that he knows the password or passphrase. For instance, knowing the PIN number will not give us access to our bank account at an ATM machine. Tokens can take many forms. One of the earliest physical tokens was a key. Today, modern tokens come in the form of a self contained hardware with computing capability. If you cannot meet the challenges of these two factors, the access will be denied.

Depending on the sensitivity of the data or the system, the number of factors for authentication may be increased. For instance, in a three factor authentication may a password, a smart card and a biometric fact such as finger print may be required.

Without a proper authentication method no other defense system will succeed. Authentication is the first line defense. After we prove that we are whom we say we are, the privileges of the user is questioned.

In the authorization stage, the system makes a decision on who can access the data or the system and what they can do with that data. As mentioned above, these privileges are summarized as read, write and execute privileges.

Clearly, authentication and authorization are closely related to each other.

II. THE SECOND LAYER: ENCRYPTION

Assuming the existence of proper logical access controls (authentication and authorization), encryption is offered as a second layer of defense for the privacy and the confidentiality of the data. While the implementations differ, the fundamentals of encryption are similar in most applications. However, it should never be considered as a substitute for access controls.

Encryption is the process of transforming information (referred to as plaintext) using an algorithm (called a cipher) to make it unreadable to anyone except those possessing special information, usually referred to as a key. The result of the process is encrypted information (ciphertext). The process of making the encrypted information readable again, is referred to as decryption.

1. How it Works

In a typical encryption process, an encryption key is used to scramble the plain text into a cipher text. Once the information is scrambled it will have the the appearance of random, unintelligible data unless you have the right key to decrypt it.

The process of confidentiality without encryption is similar to putting a single document into a vault. As long as there is a single document, the only person with that key can read that document. In the case of encryption we can have several copies of the original document outside the vault. Only the persons with the encryption key can decipher this document.

The success of encryption depends on the key management. If a key is stolen or intercepted, the encryption will provide no benefit.

Encryption keys may be classified as symmetric (a single key known by the sender and recipient) and asymmetric
key (two different keys used by the sender and recipient.). The symmetric key approach is the traditional encryption method where the sender and recipient know each other. The key is exchanged through a private channel (such a telephone or face-to-face communication).

The length of the key prevents the guessing of the key by the third parties. A strong crypto system has a large range of keystrokes so that it is not possible to try all possible combinations in a short period of time. This type of attack is called "brute force" attack. A strong crypto system should resist all known methods for breaking codes.

In a world where we deal global computer networks, exchanging symmetric keys will be impossible face to face. Or, it will be physically impossible to manage all the keys for all the users. Imagine the complication that might arise if a bank wants to send bank statements in an encrypted format. We need a separate key for each user and somebody has to maintain the privacy of these keys. Fortunately, there are other alternatives to symmetric key mechanism.

This is the environment where employ asymmetric keys. In the case of asymmetric approach, both sides may not know each other and there may not be a private channel to exchange the encryption key. The encryption problem in this environment is solved by using two different keys unrelated to each other: the public key known to everyone and the private key known only to the recipient of the cipher text. Before sending each message to the recipient, the message has to be encrypted using the recipient’s public key. Once encrypted, the data is unusable for viewing (even to the sender) or access until it is decrypted using the private key of the recipient. This framework, popular in the Internet environment, is called the Public Key Infrastructure (or PKI) and is popular in the Internet environment.

Encryption will protect the privacy of the information as it moves (physically or virtually) within a computer network or between computer networks.

The encryption for data in storage may be provided in many different levels.

As a first alternative, let us consider encrypting a database at the operating system level. This means encryption at the file level, i.e. database file as a whole. Microsoft's EFS utility (Encrypted File System) is one such tool for Windows environment. This approach brings out a serious performance problems against the database. Every read and every write is associated with encrypt and/or decrypt looks at the data as a whole. In addition to an encrypted data, numerous DBMS structures, such as indexes, have to be encrypted and decrypted along the side of data. This will create a significant performance issue and cannot last for a long time.

A more serious problem comes to surface, when we consider more than one user accessing the same database. Since there is only one encryption key at the file level, any authenticated user at the operating system level will be able to access the database as a whole. For instance, a sales manager and a salesman may use the same encryption key to access the sales data even though their information needs differ significantly. Since the encryption is handled at the operating system level, any authenticated user at that level can circumvent the objective of encryption by making copies of the database and using it in a different operating environment. For instance, a "buffer overflow attack" will easily circumvent the purpose of encryption. It is clear that, this type of encryption will not protect the data from system administrators e.g. admin or root user). A system administrator will have the read privilege to the totality of the database.

III) SOLUTION: Encrypt at the Column or Cell level

Consequently, encryption for data storage should be handled at the database level on individual’s "need to know basis" for each column and row. Users should not be allowed to access that is not related to their job descriptions. For instance, a manager in the shipping department should not have access to the accounting payroll.

The decision to encrypt should start by determining what to encrypt, which in turn depends on the sensitivity of the data. Privileges may be granted either on the use of a system (such as CREATE TABLE) or the data base objects...
(such as tables and views). The key to a successful encryption depends on data encryption in the data tables.

To further explain this concept, think of a table listing customers for a health insurer. Assume that, within this table, the following information is stored:

<table>
<thead>
<tr>
<th>CID</th>
<th>C_Name</th>
<th>C_Address</th>
<th>PolicyType</th>
<th>PreExistingCondition</th>
</tr>
</thead>
</table>

The first thing that comes to mind is to encrypt every column in the table. One should remember that this table might include a list of 10 million customers. If every column is encrypted, any data retrieval will, such as Customer ID and Customer Name will require decrypting 10 million rows. This is an unnecessary overhead for any computer system. Fortunately, an insert for a new customer will not require a substantial overhead for the system.

In this table, there is no reason to encrypt the first four fields. The most likely field to be encrypted is the field which houses the pre-existing health problems. By encrypting this field alone, the performance issues will be minimized. Any query, based on the name, address, and policy type will be responded efficiently, eliminating the overhead of encryption completely. After we find the row for a specific Customer ID, we can decrypt that specific row. The extra work load on the system will be minimized and the Select command in SQL will work faster.

Before any decision to encrypt databases, we have to study most commonly used "Select" commands used in that environment. Based on this analysis we can encrypt without affecting the system performance.

As mentioned above, key management is one of the central problems for encryption. Some implementations put the encryption key into a procedure. This approach is inherently weak if the access controls are violated. Combining access controls with encryption can be circumvented.

2. Where to use it

Keeping in mind that encryption is a supplement to access controls, no matter where they are, most sensitive data should be encrypted. We now look at the states of data in an information system. The data will appear in three distinct forms: Data in Use, Data in Transit and Data at Rest

a. Data in use is difficult to encrypt Once the data enters into a CPU- or a register, it has decrypted format.

b. The data in transit (or data in motion) usually moves back and forth between the client and the storage unit. Encryption is offered as a security tool to protect the data in both states. For instance, the sensitive data in an email should be encrypted as a general rule

   Encryption of data in transit aims to prevent the interception of the data between the client computer (or, initial storage unit) and the final storage unit. Due to dramatic increases in the volume of the Internet and wireless traffic, numerous successful security systems and protocols have been developed in recent years. SSL (Secure Socket Layer, usually identified with https://), TLS (Transport Layer Security) and IPSec (Secure Internet Protocols) are some of the popular encryption standards are sonic of the Internet based tools.

   When data is in transit, each piece of information sent by the client is encrypted as it is sent and decrypted as it is received by the recipient. If the sending and receiving units are databases, query results coming from one database to another must be encrypted at the source before it is sent and decrypted as the client receives it.

c. Data at Rest (or data at storage)

   Though there are numerous encryption methods for data in transit, we do not see the same level of serious concern for the protection of data in storage.

   Data in storage encompasses data on file servers, client hard disks, backups, and similar devices. The data security strategy should include potential risks associated with the loss or theft of data stored on small
removable media, such as flash cards, memory sticks, and optical discs. Data in storage also implies data in a Database Management System (DBMS). A standard database will typically contain a large amount of data stored in separate tables which are accessed by a wide range of users. The continual increase in the volume and complexity of these data as well as in the number of users using the system requires much greater emphasis on the security of the database objects and the access privileges.

Encryption of data in transit does nothing to protect data in storage. The database is where data is exposed to various threats for a long period of time. Many in the field might think that an authorization based on the "GRANT PRIVILEGE" of SQL, might provide the necessary protection. As we will show it below, this approach does not provide a satisfactory protection for the data.

i) The Current Practice

A study conducted by CIO magazine and PriceWaterhouseCoopers [I] has found that the stored data had a greater risk of threat than the data in motion. The 7,596 respondents of the survey indicated that most security breaches in the previous 12 months were related to the stored data. In spite of this substantial security risk, only 30 percent of IT executives said that encryption was used by their organizations to protect stored data.

In the following pages, we will show how to improve the security of data in storage.

ii) Alternative Encryption Approaches for Data in Storage

In other words, encryption should be added as another security layer. My user who has full control over the operating system will have full control over the database, and circumvent the encryption.