ISSN:0975-9646

Shivnandan Singh et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3215 - 3219

A Review Report on Security Threats on Database

Shivnandan Singh
PG Scholar, Galgotias University
Greater Noida, UP, India

Rakesh Kumar Rai
Asstt. Prof, ITS Engg College
Greater Noida, UP, India

Abstract:Data is one of the important elements for any organization. As we know that database is collection of data and programs to perform operations on that data. So for the successful run for any organization we have to secure our data. So in this paper We have to focus on threats related to database as well as several algorithms related to database security. Databases have the highest rate of breaches among all business assets, according to the 2012 Verizon Data Breach Report. Verizon reported that 96% of records breached are from databases, and the Open Security Foundation revealed that 242.6 million records were potentially compromised in 2012.

1. DATABASE SECURITY THREATS:

There are many ways of securing the database. These ways are based on different aspects of securing the database. Different aspects with traditional approaches are summarized below.

1.1 - Excessive Privilege Abuse When end users or applications are granted database these privileges may be abused for malicious purpose access privileges that exceed the requirements of their job function. For example, a university administrator whose job requires only the ability to change student contact information may take advantage of excessive database update privileges to change marks.

A given database user ends up with excessive privileges for the simple reason that database administrators do not have the time to define and update granular access privilege control mechanisms for each user. As a result, all users or large groups of users are granted generic default access privileges that far exceed specific job requirements.

1.2 -Preventing Excessive Privilege Abuse - Query-Level Access Control The solution to excessive privileges is query-level access control.

The granularity of data access control must extend beyond the table to specific rows and columns within in table. Query-level access control refers to a mechanism that restricts database privileges to minimum-required SQL operations (SELECT, UPDATE, etc.) and data. A sufficiently granular query-level access control mechanism would allow the rogue university administrator described previously to update contact information, but issue an alert if he attempts to changes grades. Query-level access control is useful not only for detecting excessive privilege abuse by malicious employees.

1.3 - Legitimate Privilege Abuse Users may also abuse legitimate database privileges for unauthorized purposes. Consider a hypothetical rogue healthcare worker with privileges to view individual patient records via a custom Web application. The structure of the Web application normally limits users to viewing an individual patient's healthcare history – multiple records cannot be viewed simultaneously and electronic copies are not allowed. However, the rogue worker may circumvent these limitations by connecting to the database using an alternative client such as MS-Excel. Using MS-Excel and his legitimate login credentials, the worker may retrieve and save all patient records.

It is unlikely that such personal copies of patient record databases comply with any healthcare organization's patient data protection policies. There are two risks to consider. The first is the rogue worker who is willing to trade patient records for money. The second (and perhaps more common) is the negligent employee that retrieves and stores large amounts of information to their client machine for legitimate work purposes.

Once the data exists on an endpoint machine, it becomes vulnerable to, Trojans, laptop theft, etc.

1.4 Privilege Elevation Attackers may take advantage of database platform software vulnerabilities to convert access privileges from those of an ordinary user to those of an administrator. Vulnerabilities may be found in, built-in functions, stored procedures, protocol implementations, and even SQL statements. For example, a software developer at a financial institution might take advantage of a vulnerable function to gain the database administrative privilege. With administrative privilege, the rogue developer may turn off audit mechanisms, transfer funds, create bogus accounts, etc.

Preventing Privilege Elevation – IPS and Query Level Access Control Privilege elevation exploits can be prevented with a combination of traditional intrusion prevention systems (IPS) and query-level access control (see Excessive Privileges above). IPS inspects database traffic to identify patterns which correspond to known vulnerabilities. For example, if a given function is known to be vulnerable, then an IPS may either block all access to the vulnerable procedure, or (if possible) block only those procedures with embedded attacks.

1.5: Platform Vulnerabilities in underlying operating systems (Windows 2000, UNIX, etc.) and additional services installed on a database server may lead to unauthorized access, data corruption, or denial of service. The Blaster Worm, for example, took advantage of a Windows 2000 vulnerability to create denial of service conditions.

Preventing Platform Attacks - Software Updates and Intrusion Prevention Protection of database assets from platform attacks requires a combination of regular software updates (patches) and Intrusion Prevention Systems (IPS).

Vendor provided updates eliminate vulnerabilities found in database platform over time. Unfortunately, software updates are provided and implemented by enterprises according to periodic cycles. In between update cycles, databases are not protected. In addition, compatibility problems sometimes prevent software updates altogether. To address these problems, IPS should be implemented. As described previously, IPS inspects database traffic and identifies attacks targeting known vulnerabilities.

1.6 - SQL Injection In a SQL injection attack, a perpetrator typically inserts (or "injects") unauthorized database statements into a vulnerable SQL data channel. Typically targeted data channels include stored procedures and Web application input parameters. These injected statements are then passed to the database where they are executed. Using SQL injection, attackers may gain unrestricted access to an entire database.

Preventing SQL Injection Three techniques can be combined to effectively combat SQL injection: intrusion prevention (IPS), query-level access control (see Excessive Privilege Abuse), and event correlation. However, IPS alone is not reliable since SQL injection strings are prone to false positives. Security managers who rely on IPS alone would be bombarded with "possible" SQL injection alerts. IPS can identify vulnerable stored procedures or SQL injection strings. However, by correlating a SQL injection signature with another violation such as a query-level access control violation, a real attack can be identified with extreme accuracy. It's unlikely that a SQL injection signature and another violation would appear in the same request during normal business operation.

1.7 - Weak Audit Trail Automated recording of all sensitive and/or unusual database transactions should be part of the foundation

Underlying any database deployment. Weak database audit policy represents a serious organizational risk on many levels.

• Regulatory Risk - Organizations with weak (or sometimes non-existent) database audit mechanisms will increasingly find that they are at odds with government regulatory requirements. Sarbanes-Oxley (SOX) in the financial services sector and the Healthcare Information Portability and Accountability

Act (HIPAA) in the healthcare sector are just two examples of government regulation with clear database audit requirements.

• Deterrence – Like video cameras recording the faces of individuals entering a bank, database audit mechanisms serves to deter attackers who know that database audit tracking provide investigators with forensics link intruders to a crime.

• Detection and Recovery – Audit mechanisms represent the last line of database defense. If an attacker manages to circumvent other defenses, audit data can identify the existence of a violation after the fact. Audit data may then be used to link a violation to a particular user and/or repair the system.

1.8 - Denial of Service Denial of Service (DOS) is a general attack category in which access to network applications or data is denied to intended users. Denial of service (DOS) conditions may be created via many techniques - many of which are related to previously mentioned vulnerabilities. For example, DOS may be achieved by taking advantage of a database platform vulnerability to crash a server. Other common DOS techniques include data corruption, network flooding, and server resource overload (memory, CPU, etc.). Resource overload is particularly common in database environments.

The motivations behind DOS are similarly diverse. DOS attacks are often linked to extortion scams in which a remote attacker will repeatedly crash servers until the victim deposits funds to an international bank account. Alternatively, DOS may be traced to a worm infection. Whatever the source, DOS represents a serious threat for many organizations.

1.9 Database Communications Protocol Vulnerabilities A growing number of security vulnerabilities are being identified in the database communication protocols of all database vendors. Four out of seven security fixes in the two most recent IBM DB2 FixPacks address protocol vulnerabilities1. Similarly, 11 out of 23 database vulnerabilities fixed in the most recent Oracle quarterly patch relate to protocols. Fraudulent activity targeting these vulnerabilities can range from

unauthorized data access, to data corruption, to denial of service. The SQL Slammer2 worm, for example, took advantage of a flaw in the Microsoft SQL Server protocol to force denial of service. To make matters worse, no

record of these fraud vectors will exist in the native audit trail since protocol operations are not covered by native database audit mechanisms.

Preventing Database Communication Protocol Attacks Database communication protocol attacks can be defeated with technology commonly referred to as protocol

Validation. Protocol validation technology essentially parses (disassembles) database traffic and compares it to expectations. In the event that live traffic does not match expectations, alerts or blocking actions may be taken.

Secure Sphere's Database Communication Protocol Validation audits and protects against protocol threats by comparing live database communications protocols to expected protocol structures. No other database security or audit solution provides this capability. It is derived through the Imperva Application Defense

1.10: Weak authentication schemes allow attackers to assume the identity of legitimate database users by stealing or otherwise obtaining login credentials. An attacker may employ any number of strategies to

obtain credentials.

• Brute Force - The attacker repeatedly enters username/password combinations until he finds one that works. The brute force process may involve simple guesswork or systematic enumeration of all possible

username/password combinations. Often an attacker will use automated programs to
accelerate the brute force process.
• Social Engineering – A scheme in which the attacker takes advantage the natural human tendency to trust in order to convince others to provide their login credentials. For example, an attacker may present himself via phone as an IT manager and request login credentials for "system maintenance" purposes.
• Direct Credential Theft – An attacker may steal login credentials by copying post-it notes, password files, etcCenter's (ADC) ongoing research into proprietary database communication protocols and vulnerabilities.
Database and application vendors including Oracle, Microsoft, and IBM have credited the ADC with the discovery of serious vulnerabilities and mitigation techniques that have led to increased security in their products. Based upon this research, Imperva is able to incorporate unmatched protocol knowledge into Secure Sphere.

1.11: Backup database storage media is often completely unprotected from attack. As a result, several high profile security breaches have involved theft of database backup tapes and hard disks. All database backups should be encrypted. In fact, some vendors have suggested that future DBMS products may not support the creation of unencrypted backups. Encryption of on-line production database information is often suggested, but performance and cryptographic key management drawbacks often make this impractical and are generally acknowledged to be a poor substitute for granular privilege controls described above.

2. DATABASE SECURITY TECHNIQUES

To implement any security solution for a computing system, organization must ensure three aspects namely policy, mechanism and assurance. The requirements that must be implemented in hardware, software and outside the computing system are defined under policy. Proper mechanism to implement the requirements discussed in the policy is indispensable for an organization to have good computing security solution. Assurance is to ensure that mechanism meets the policy requirements of the organization [1].
Different users have different access rights on different database objects. Access Control Mechanisms deal with managing the access rights. It is the basic technique to protect the data objects in the databases and is supported by most of the DBMS [2]. Then some techniques to fight with SQLIA are discussed in the paper.

A. Access Control Mechanisms

Access Control Mechanism is a technique to maintain data confidentiality. When someone tries to access data object, Access Control Mechanism checks the rights of the user against set of authorizations. They are generally specified by security administrator or security officer. Authorizations are given as per the security policy of the organization. Along with Access Control Mechanism, A strong Authentication mechanism is also required to authenticate the valid user of a database system. After that access control will help defining different access permissions on different data objects of a database [2].
There are some proposed models for access control in relational database systems. These models give approaches for implementing access control in databases.
1) Discretionary Access Control Models: In this approach, depending on user's identity and authorization rules, access is given to data objects as per some discretionary policies. Main advantage here is users can grant authorization on data objects to other users. Because of such flexibility, this is widely used technique in many organizations. Active entities like users in security system are also called subjects.
Authorization administration is the function of granting and revoking authorizations. So by authorization administration, authorizations are entered into or removed from access control mechanism. There are two main types of such administrations Centralized administration is the type of administration in which some privileged subjects or users can grant or revoke authorizations. Ownership administration is the one in which creator of the object grants and revokes access to object. Owner can give other users right to grant or revoke some or all the access on the object in ownership administration.
2. Content based access control: In this model, access control decisions should be based on contents of the data[6]. E.g. Customer table has information of all the customers of company. So only those employees, who are related to projects about those customers, should have access to those customers. This approach is generally implemented using views. Protection views are used to support content based access control. Shorthand views are used to simplify query writing. Access policies here can be expressed in high level language. Modifications performed in the data do not require changes in the access control policies. If new data is inserted which satisfies some policy, then it will be automatically as part of data returned by corresponding view [2].
3.Fine grained access control: This mechanism supports access control at tuple level [2]. Therefore the name is Fine grained access control. It allows granular level of access control. To implement such a scheme, it requires specialization of views. Oracle virtual private Database and Truman model are the examples implementing such access mechanism [2].
4) Mandatory Access Control Model: Mandatory access control is based on the classification of data objects and users [5][6]. Classification is based on the partially ordered set of classes called access classes.
Access class contains Security level and set of categories. Security level represents sensitivity of information [5][6]. Access control in this access model is based on following two principles. No read-up: User can read only those data objects whose access classes are dominated by access class of user. No write-down: User can write only those data objects whose access classes dominate access classes of the user.

These principles restrict the flow of sensitive data into data objects at lower or incomparable access classes [6].There are advanced database concepts like object relational and object based databases. These access control mechanisms are modified to suite these databases. Even some models are proposed for XML data [5][6].

B. Techniques to fight with SQLIA

As discussed earlier, SQLIA is the most dangerous attack on databases. This section discusses some of the techniques to detect and prevent SQLIA. The detection approaches for SQLIA can be categorized broadly into pre-generated and post-generated approaches. Post-generated approaches are generally useful while analyzing dynamic SQL which is generated by web application. Pre-generated approaches are generally used during the testing phase of the web application. Some post generated approaches are [3]:

Positive tainting and Syntax aware evaluation: Here, valid input strings are initially provided to system for detection of SQLIA. It categorizes input strings and propagates non trusted strings on fly. Syntax aware evaluation is done on the propagated strings in order to decide the non trusted strings. Syntax evaluation is performed at the database interaction point. There are some issues with this method. Initialization of trusted strings depends on developer and storage of trusted strings may lead to another attack.

Context Sensitive String Evaluation: Here any user given data are considered as non-trusted and application given data are considered as trusted. Non-trusted metadata is used for syntax analysis. This syntax analysis differentiates string and numeric constants. Then all the unsafe characters are removed from alpha numeric identifiers. Some issues in this approach are initialization of unsafe characters is developer dependent and removal of unsafe characters restricts application functionality.

Parse tree evaluation based on grammar: In this approach, SQL queries generated from user input is parsed using pre-defined grammar.

C. Data Encryption

This is the basic technique used for securing any kind of information or data. So this technique can even be applied to databases.

Figure: 1

While performing database encryption, a decision about whether to perform the encryption inside or outside the database must be taken. Some of the issues involved in this technique are How to secure keys from attacker of the system? How to give administrative rights of manipulating data using keys? And How to provide limited access for keys?

The important aspects which need to be considered while encrypting database is how to manage the encryption keys. Some of the aspects related to this issue are Number of encryption keys required, storage of keys, protection for the access of keys, and frequency of change of keys recommended approach for storing the keys is, separate the keys and data residing in the database. Generally the keys are stored in hardware like access restricted files or hardware storage modules. The process of encryption can be performed either within the database or outside the database. But there are performance and security tradeoffs which need to be considered while implementing this policy. If encryption is performed within the database, then there is less impact on application environment. Understanding the encryption algorithm supported by DBMS also plays key role while devising strategy to implement this technique. The drawback of this approach is encryption keys also are stored in the same database.

Another way to implement encryption in database is performing it on separate encryption servers. Encryption and decryption computations are performed encryption server. So here overhead of encryption is removed from DBMS and moved on to separate encryption servers to maintain the performance of DBMS. Encryption keys and data can also be separated. This approach is usually followed while encrypting database [4].The algorithms which are generally used for database encryption and often supported by DBMS are DES, Triple DES, RC2, RC4, DESX and AES.

There are various configurations available for encrypting and decrypting databases. Some of them are listed below [2].

File System Encryption: Here the physical disk where database resides is encrypted. Entire database is encrypted using single encryption key so discretionary access control cannot be implemented.

DBMS Level Encryption: There are many schemes for this kind of encryption. One scheme is based on Chinese Remainder theorem in which every row is encrypted using different sub keys for different cells. So encryption at row level and decryption at cell or field level is possible by this scheme.

There are some schemes based on Newton's interpolation polynomials which are used for database encryption.

There is a SPDE scheme which encrypts each cell I the database with its cell coordinates like table name, column name and row id etc. So in this scheme static leakage attacks and splicing attacks are prevented.

Application level Encryption: In this technique, a middleware is suggested which translates queries fired by user into new bunch of queries which will execute on encrypted database.

This technique was implemented in Data Protector System.

Client-side encryption*:* This technique is generally used in case of ―Database as a service‖ scenario where the entire database is outsourced by the organization to reduce the maintenance costs. So here data privacy is the major concern. Encryption is the basic solution in this scenario.

Indexing encrypted data*:* There are many indexing mechanisms proposed. B tree index structure is prepared over plain text values in the table and then encryption of the table is performed at the row level. Encryption of the B-tree is done at the node level.

D. Data Scrambling

Data Scrambling is a process of making sensitive information in non-production databases safe for wider visibility [5][6].Data scrambling is also known as data sanitization, data masking and data obfuscation.

Data scrambling is generally used when users have proper access to data in the database but still it is required to secure sensitive information from them. Examples of such users can be third party developers or testers working on data in database. So the values of the data which are sensitive are changed but still the values are realistic in nature.

E. Miscellaneous techniques for database security

To avoid inference problems in databases, poly instantiation technique can be used. In this method, multiple instances of the same data are allowed but the classification of such data should be different. It has effect on relations, tuples and data elements. The relations with poly instantiation are relations with different access classes. There are two types of poly instantiations namely visible and invisible poly instantiations. When a higher user inserts data in field having low data, the high data is entered as new tuple in visible poly instantiation. When low user inserts data in field that already has high data, low data is entered as new tuple. This is called invisible poly instantiation [7].

Auditing is another technique which can be used to fight with inference attack on databases. A history of all the queries fired by users is maintained in this approach. Analysis of such queries can help in finding inference attack and can be avoided in later stage. But this approach detects very limited inference attack instances and difficult to implement practically [7].

CONCLUSION

Databases form the backbone of many applications today. They are the primary form of storage for many organizations. So the attacks on databases are also increasing as they are very dangerous form of attack. They reveal key or important data to the attacker. Various attacks on databases are discussed in this paper. Review of some important database security techniques like access control, techniques against SQLIA, encryption and data scrambling are discussed. Even some future research areas in the field of database security are also discussed in this paper. This research will lead to more concrete solution for database security issue.

REFERENCES

[1] SushilJajodia, ―Database security and Privacy‖, ACM Computing Surveys, Vol. 28, No.1, March 1996

[2] Elisa Bertino, Ravi sandhu, ―Database Security- Concepts, Approaches and Challenges, IEEE Transactions on Dependable and Secure Computing, Vol 2, No 1, January-March 2005

[3] Debasish Das, Utpal Sharma, D.K. Bhattacharyya, ―An Approach to detection of SQL Injection Attack Based on Dynamic Query Matching, International Journal of Computer Applications (0975-8887),vol 1 –No 25,page no 28-34.

[4] RSA Security Inc., **"**Securing Data at Rest: Developing a Database Encryption Strategy, A White Paper for developers, e-business managers and IT‖, Website, September 7 2012, http://www.rsa.com/products/bsafe/whitepapers/DDES_WP_0702.pdf

[5] A NET 2000 Ltd., **"**Data Scrambling Issues‖, A White Paper(2010), Website, October 10 2012, http://www.datamasker.com/datascramblingissues.pdf

[6] Huw Price, **"**A Short Guide to Scrambling, Masking and Obfuscating Production Data, Grid Tools‖ White Paper, Website, October 15 2012, http://www.grid-tools.com/download/Data_Masking.pdf

[7] Ravi S Sandhu, SushilJajodia, ―Data and database security and controls‖, Handbook of security management ,Auerbach publisher, 1993, pages 481-499