University of Thi-Qar Journal Vol.10 No.2 June 2015
Web Site: https://jutq.utq.edu.iq/index.php/main *Email: journal@jutq.utq.edu.iq*

Databases Attacks and Defence Techniques

https://doi.org/10.32792/utq/utj/vol10/2/3

Ali Ghalib Mohammed Khidhir *

*Physics department, Science College, Thi-Qar University

ABSTRACT

In today's world, any and every organisation keeps data. And the most common and widely used method of storing data is databases. It allows those in possession an easy way to retrieve and maintain the information as well as organise and manipulate the data when necessary. These jobs are carried out using Database Management Systems. Keeping data confidential is one, if not the, most important task of organisations, so it is essential to find a method or methods which keep databases secure. This paper will review in detail the main database security techniques and outline to some extent the ways in which databases are attacked.

1. INTRODUCTION:

Databases are the primary means of storing data in all businesses and organisations. Some organisations would collapse if something were to happen to their databases, which is why much emphasis is placed on securing the information stored on them. The organisations success or failure depends on their databases and cannot afford to have something happen to them [1].
Databases are usually structured with information generally stored in relational databases. This is how the majority of the data will be organised. This mode of storage puts database information into table format. Each table contains the information which relate to one another. This is the most common mode of storage used by organisations [2].
Common database security is divided into three categories: *DBMS security, physical security* and *operating system security*.These

systems have been useful with regard to access control as long as it is accessed in the intended ways. But if an attacker gained access by bypassing the intended access routes, such as by getting information from insiders who preside over these databases (such as system or database administrators), then these security measures would not suffice. Therefore to have just the aforementioned layers of security is not enough to guarantee the security of the data.

Further security measures must be implemented to ensure the safety of any data. There are advanced methods used today by organisations to ensure their databases are well secured, which is encrypting the data. This is used in a wide range of organisations, especially banking, government, health care and finance industries. However this does not prevent from all kinds of attack [3].

The security of computer databases addresses three key aspects, which are; *Confidentiality*, *Integrity* and *Availability*. With regards to confidentiality, the security system must ensure that all data is only accessible by authorized parties. Protecting the integrity of the data ensures that only those authorised can carry out any modifications to the data stored. And availability ensures that it can be accessed whenever necessary by any authorised party.

Database security has certain requirements that it must fulfil, such as physical, logical and elemental integrity. Also it must include access control, user authentication and availability measures as well as the ability to be audited. With regards to physical database integrity, this deals with physical problems such as power cuts etc. Logical database integrity relates to the preservation and maintenance of the structure and relations within a database. Element integrity ensures the accuracy of the data stored. Access control limits the user to only view data they are authorised to. Authentication and availability handles verification of the user credentials and making available the data the user wants to view. And auditing relates to the tracking of any changes made to the database and any users who have accessed it [4].

2. DATABASE MANAGEMENT SYSTEMS (DBMS)

These are applications which are used in managing the data on databases [4]. It assists those managing and using the data by organising the data, making it easier to retrieve information and perform more effectively. It also keeps logs of transactions which help in data recovery.
Databases are the backbone of most organisations, so it is essential that the data they contain are secure. Should their information be leaked to the outside world, become damaged or altered in any way, this could spell the end for that organisation. Therefore, a brief description of the different ways in which databases may be attacked will be given before the security techniques, which are available, are discussed.

3. TYPES OF ATTACKS ON DATABASES:

Databases can face various modes of attack in today's world. The most significant and common attacks will come under the umbrella of four methods: SQLIA, Inference, Passive Attacks and Active Attacks.

I. Passive Attacks:

A passive attack is when the attacker observes the present data on the database only. This type of attack can be carried out in three ways [4].
Dynamic leakage-This attack entails changes being carried out over a period of time, during which it can be observed and plain text value information can be extracted.
Static leakage -With this attack type, a snapshot of the data is taken through which information is obtained.

Linkage leakage- This type of attack gives the attacker information about plain text values through linking table valueplaces to those values in index [4].

II. Active Attacks:

In the case of active attacks, the table values themselves are altered. This can be a devastating method of attack and can be carried out in various ways;
Spoofing- This method is carried out by placing a generated value with a cipher value. An attacker might even try to replace a current cipher value in the place of another. This type of attack poses a relatively low risk if encryption keys are secure.
Splicing -In this type of attack, cipher text values are replaced by different cipher text values. The attack would take encrypted text from one location and copy it to the location which is under attack.
Replay- These attacks are carried out by replacing a cipher text with a previously used or deleted version.

III. *Inference:*

This is a major attack on databases. This mode of attack is a way of deriving sensitive data from non sensitive data [3]. How this attack is launched is by firing a query which is specific in nature which matches one exact data item. Indirect attacks on databases under inference include using statistical data to gain sensitive information. This attack may be launched in different ways. One way is the attacker would try to infer information from sum taken from values used in reports. Count function can be used along with the Sum function to gain sensitive information. Both functions are commonplace in DBMS as default functions. Attackers can also use medians as a statistical measure to determine genuine sensitive data, although this method is a difficult one. Another possible inference attack is called a 'Tracker' attack. With this kind of attack a desired data can be obtained using additional queries which produce small results. The attacker would then request additional records to be retrieved for two different queries which would, in turn, have them cancel each other out, leaving only the desired data which would then be collected. One of the more common types of tracker attacks is called a 'linear system vulnerability' method. It requires the use of logic and algebra to find the data distribution function through which, one can locate those desired elements. A series of query sets are used instead of two opposing query sets, again leaving the attacker the desired information [3].

IV. *SQLIA (SQL Injection Attack)*

This is one of the major types of attacks on databases. Most web applications use SQL queries without applying proper user validation. And this is the basic reason for SQLIA. The attackers make the server run malicious SQL queries and therefore are able to manipulate the database [5].There are five types of SQLIA:

1. *Bypassing web authentication* - Here, the attacker uses the Input Field which is where queries are input. For example, an attacker would write a query like, "Select name from User_tbl where User_nm=' ' or 1=1- -' and pwd=' ' ."Because of - - text after that will be comments ad 1=1 is always true so the user will be logged with the privilege of the first user stored in User_tbl[6].
2. *Injection with the union query* - This type of attack is when then attacker gets data from a table which is different from the one that was intended by the developer. For example, Select * from User_tbl where User_nm=' ' UNION select address from User_dtl where user_id=123-- and pwd='1t@d'. Therefore, user_123 address will be displayed [6].

3. *Damaging with additional injection query*- In this type of attack, an attacker would enter data such as an additional query along with the original query. An example of this type of attack can be the use of queries such as drop table, table_nm or delete from table_nm. Putting this through the Input Fields will generate these queries along with the original query [6].
4. *Database fingerprinting* -In this type of attack, an attacker would generate logically incorrect or illegalqueries. This would, in turn, causes the database to put out error messages which contain names of database objects such as table names, views, stored procedures and so on. And from such error messages that are put out, the attacker can derive the database used by the application as the errors put out by databases vary from one to another [6].
5. *Remote Execution of stored procedures*-This attack entails executing some stored procedures which could have harmful effects on the database.

In addition to these forms of attacks, there is also the legitimate threat of a user who has been entrusted with access, to abuse the trust given to them and expose any sensitive information to attackers or other sources, either within the organisation or externally. This kind of attack is especially dangerous for organisations such as the Ministry of Defence as data they have can cause a wide range of problems, for them and the world as a whole. Other firms that could be highly affected by such leaks include banking and finance organisations as well as some commercial organisations as they all store vast amounts of sensitive information. Leaks could harm organisations profits or the customers they serve, if not both simultaneously. The following section will describe in detail the database security techniques widely used to combat such intrusionsand avoid any type of attacks which can be launched at them.

4. DATABASE SECURITY TECHNIQUES

When applying any type of security solution to a computer system, there are three criteria that must be met. These are: *Policy*, *Assurance*, and *Mechanism*. The requirements that are defined under ***policy*** must be implemented in hardware, software and outside the computing system. ***Assurance*** relates to the ensuring of the mechanism meeting the required specifications of the organisations policy. ***Mechanism*** deals with implementing the requirements defined in the policy which is essential for an organisation's successful computer security solution [7].

5. ACCESS CONTROL MECHANISMS

One of the key mechanisms within the security system is ***access control***. This deals with authorising different user's access rights to different database objects. It is basically a technique to protect sensitive data in databases and is supported by most DBMS [1].

the *Access Control Mechanism* is one of the most common security techniques. It basically checks if users are authorised to view the data they request. Generally they are specified by the security administrators. It works alongside an authentication mechanism which validates user's requests to access different databases objects [1].

There are several different types of access control models in a relational database system. Each model has a different approach to implementing access control.

1. *Discretionary Access Control Models* -This approach gives users access to data objects depending on their identification and authorisation, as per some discretionary policies. The advantage of this model is that users here can grant authorisation to others to access certain data objects. Due to its flexibility, this is the most widely used model by organisations. In such a model active users are often referred to as subjects. The authorisation administration is the ones responsible for granting and revoking authorisation to access databases. Such administrations are split into two main categories;
    A. Centralised administration: This type of administration gives users or subjects the authorisation to grant others authorisation, as well as revoke them.
    B. Ownership administration: In this type, only the owner of the objects of the database can give authorisation to users or subjects and only they can revoke such access [7][1].
2. *System R authorisation model* - Data objects protected under this form of authorisation model are views and tables. Access modes under this model are usually limited to a few, such as Select, Insert, Update, and Delete. In today's environment, this model is extended to include data objects like triggers as well. The one who created the table is the owner; therefore they are the ones who grant subjects access and usage options. There are some extensions which have been introduced to this model such as negative and positive authorisations, which were implemented due to the nature of the model and the restrictions that came with it. Those additions reduced restrictions without compromising the system integrity [1].
3. *Content Based Access Control*-This model dictates access control decisions be based on the contents of the data. For example, if the data contains information pertaining to customers whom the employees have projects involving them, then they are granted authorisations to those customers' data specifically. Generally, this approach is applied using views. Views protection is used to aid content based access control. Shorthand views are used to simplify query writing. Also the data does not require changes in access control policies if there have been any modifications made. If any new data inserted meets the criteria of the policy, it will automatically be returned by the corresponding view [1].
4. *RBAC models*-RBAC stands for Role Based Access Control. It is based on the roles of the users within the organisation. Authorisation is given to those who are in the relevant roles or jobs which require access to the information on the database. RBAC is a very simple model. When a user changes their role in the organisation only the access authorisation with the previous function needs to be revoked, making the change simple and straightforward.
5. *Mandatory Access Control Model*-Mandatory access control is based on the classification of data objects and users [1][7]. There are structured classes called access classes. Each access class is given a security level and set of categories. The security levels represent the degree of sensitivity of the information contained on the database. This model of access control is based on two principles. One is called 'No read-up' in which a user can only read data objects whose access classes are dominated by the access class of the

user. The other is called 'No write-down' in which the user can only write data objects whose class are dominated by the access classes of the user. These principles restrict the flow of sensitive data into data objects of a lower access class [1].

6. Techniques used to combat SQLIA

With SQLIA being the most significant and dangerous threat to databases, a detailed discussion on the security systems and methods used to fight and detect such attacks will be covered in the following section.

With regards to the approach taken in detecting SQLIA, it can be categorised into two main parts, namely pre-generation and post-generation. Pre-generated approaches are usually used during the testing phase of any web application while the post-generated approach is used while analysing dynamic SQL which is generated by web applications [6].

I. Post-generated approaches:

Positive tainting and Syntax Aware Evaluation - With this technique, valid input strings are initially provided to the system for the detection of SQLIA. It then categorises input strings and propagates all no trusted strings on fly. Syntax aware evaluation is then carried out on those propagated strings to find out which of the strings are non-trusted. This evaluation is carried out at the database interaction point. Although this method is useful, there are some problems as the initialisation of trusted strings depends on the developer and the storage of those strings may lead to another attack.

Context Sensitive String Evaluation-This technique considers any user given data as non-trusted, while any application given data are considered trusted. Non-trusted data is then used for syntax analysis. Syntax analysis then differentiates sting content from numeric content. Then it removes all unsafe characters from alpha numeric identifiers. One negative feature of this approach is that the initialisation of unsafe characters is developer dependent, which results in the restriction of the applications functionality.Special literals are used to mark each ends of a strings. But a drawback with this is that an attacker is able to manipulate input sting by using different kinds of these special characters.

II. Pre-generated approaches:

Pixy - This is a static analysis tool which is used to find any vulnerabilities in web applications. It does this by using data flow analysis to form statistical information for each programme point. To identify points where malicious data can enter applications, parse trees are developed and taint analysis tools are used. One negative point about this method is that an attacker can explore and find the vulnerabilities of this tool as it is an open source.

Programme Query Language - This web language has pre-defined grammar and is the language of web developers to detect attack related queries. It works by having a translation made from PQL to data logs which then help the programmes provide support in finding malicious queries. The detection of such malicious queries depends on the developer input which is in the data used [6].

There are also some new techniques which have been developed to detect SQLIA. One of which is called DUD which stands for Debasish, Utpal and D.K. Bhattacharya.

How it works is it has some user defined threshold, which lets say it is called e. It also has SQL Master File which contains a list of all legitimate queries. Approximate Matching () algorithm computes the differences between legal queries in SQLMF and XSQL queries and compares it to that in threshold e. Those queries within the threshold will be allowed to pass and those which aren't will not pass the database server [6].

One benefit of this approach is it avoids initialisation of trusted as well as non-trusted strings. Another is that matching logic in DUD is easy to do and SQLMF can be updated. It is also developer independent [6].

III. Data Encryption:

The most common technique used by businesses and organisations for safeguarding sensitive information is data encryption. It is a basic but secure technique which can be applied in many ways, including on databases. There are several types of encryption methods, some of which will be described in the following section.
Handling encrypted data can add to the time it takes to carry out a task, so to limit this, certain criteria for data which should be encrypted should be met:
(1) Only sensitive data should be encrypted while non-sensitive data should be left unencrypted [8].
(2) Only data of interest should be encrypted or decrypted when queries are executed [8].

(3) It is desired that the encrypted database not require much more storage than the original [8].

IV. File system Encryption:

This scheme suggests that the entire physical disc be encrypted which would ensure the protection of the database. However, this comes with a drawback as using this scheme would mean that the entire database is protected by a single encryption key, thus discretionary access control could not be supported [9].

V. DBMS-Level Encryption:

This type of encryption can be implemented in various ways. One way is called the Chinese-Reminder theorem, in which each row is encrypted using different sub-keys for different cells. Using this scheme allows encryption at the level of rows and decryption at the level of cells. Another scheme extends on the capabilities of the aforementioned scheme by supporting multilayer access control. Subjects and objects are classified into distinct security classes that are ordered in a hierarchy, which means that an object with a certain security class can be accessed by subjects in the same class or higher [10][11].

Another scheme is based on the Newton's interpolating polynomials system of database encryption. A more common encryption scheme is the RSA public-key system which has two database encryption schemes. One is single column orientated and the other is single row orientated. A disadvantage of this scheme is that the basic element in the database is a row and not a cell, which means the database structure will be modified. Another disadvantage which applies to all the above mentioned schemes is that if one cell were to be modified then the whole row would require re-

encryption, which would mean that all encryption keys need to be readily available [11].

A scheme which does not have this problem is the SPDE scheme which encrypts each cell in the database individually along with its cell co-ordinates which includes the table and column names and row-ID number. With this method, static leakage attacks are prevented since equal plaintext values are encrypted to different cipher-text values. Splicing attacks are also prevented by this security scheme, as each cipher text is correlated to a specific place which means trying to move it to a different location can easily be detected [13][12].

VI. Web Application Encryption:

This data protection scheme is known as WDSP which stands for Web Data Service Provider Middleware. This application translates user queries into a new set of queries which are executed off the encrypted DBMS. This model basically acts as a middleman which regulates access to secure data stored on the web service provider. This mode of security has become quite popular on the web these days due to the protection it provides to public data storages, back-up data and sharing services.

VII. Client-Side Encryption:

Due to the increased number of people using the internet, together with advances in software and networking, organisations are now able to share data in a variety of ways with ease. This has led to a new paradigm named "Database as a Server" (DAS), through which all of the data management process is outsourced by organisations which would reduce their costs and allow them to focus more on the core of the business [15][14].

One drawback of this scheme is the performance degradation due to the remote access to data. Another is data privacy, as sensitive data will have to be stored securely, and protected against untrustworthy servers. To define this encryption system under the assumption that the server is not trusted raises the question of how a query can be evaluated if data are encrypted and the server has no access to the encryption keys [16].

VIII. Indexing Encrypted Data:

There are various indexing mechanisms available, one of which involves making plain text values, followed by encrypting each page of the index individually. A modification to this which is suggested is encrypting different index pages with different keys depending on the number of pages. A more popular mechanism under this umbrella is the 'B tree' structure. A B tree index structure is encrypted at the node level. This structure is prepared over plain text values in a table and then the encryption of the table is carried out at row level [23].

IX. Keys Management:

Many encryption techniques have been mentioned in this paper however, most of them are not very flexible nor convenient in application. The Keys Management scheme fills that void with its ability to not only keep data safe but allow for easy access and sharing also. What it does is generate two separate pairs of keys, one of

which is kept by the user at the client end and the other (the public key) is kept in the database server [17][18].

X. Data Scrambling:

Data scrambling is known by many names including data masking and sanitization. It involves making sensitive data in non-productive databases safe for wider visibility [19][20]. It is generally used to protect extra sensitive information from users who have access to the database, such as testers working on the database and third party developers. What data scrambling does is change the values in the database but keeping them realistic in nature.

There are several different methods of data scrambling. One is to create a set of views which can be used to mask the data and create a secure environment. Another method is to make a copy of the production data, update data and then copy it to development to complete the task [21]. One benefit of data scrambling is its traceability, which is useful if data is lost (data recovery). Other scrambling methods include using database functions built-in the DBMS and using seed tables which is an effective technique amongst others.

Although scrambling data is an effective method of securing data, there are some associated issues. For example, the scrambled data has to resemble original data. The contents in one column in a row must resemble the contents in another column in the same row. They should also have the same relationship to other values in the columns/rows. Also, data which has been scrambled can be used as a join key to other tables so that scrambled data in one table must be synchronised with data changes in several other tables. Care must also be taken when scrambling data to ensure that the scrambled value should not exceed previously defined limits as well as maintaining consistency across data across all tables [19][20].

Encryption can be used instead of scrambling in some cases, but in larger databases and data warehouses, encryption carries an overhead on the system. It involves added costs and response times for queries will be increased [22]. A technique known as MOBAT which stands for Modulus Based Technique is used for these types of larger scale encryptions. It carries out encryption in three stages. The first stage involves the user application itself. The second stage is using modulus based technique to mask the third part which is the database itself.

Before data scrambling be applied to any database, certain things must be known which include documentation of the database, the relationships among the tables within the database and the users knowledge of the key and sensitive data as well as naming standards used in the organisation [21].

7. CONCLUSION:

There have been many points mentioned already emphasizing the importance of keeping databases secure and the value the hold for an organisation. Their safety is paramount as it is the primary form of storage for most organisations today. Databases are the backbone of most applications so any attacks made on them are very dangerous and pose substantial risks to the welfare of the business or organisation.

There are a lot of improvements which could be made to database security techniques. IBM's research centre SPIDER (Self Protecting Database Research) are working on

autonomic database security, steps are being taken towards making database security even safer.
Based on an analysis of the different security schemes available, the method which offers the best flexibility is when encryption is made inside the DBMS. Using the File System approach may be easy to deploy, comes with restrictions such as not being able to use different encryption keys. It also does not allow someone to chose which data can be encrypted and decrypted, which has a significant influence on both the way the database performs and the security of the data.

All aforementioned techniques have positive and negatives but with encryption researches and developers focusing on departments such as unauthorised leakages and modifications, intrusion detection and query logs amongst others, encryption techniques will give better guarantees of the safety of databases.
The database industry is growing and new approaches are being created and improvements to existing techniques are being made. The area of cloud computing which involves resources including databases are shared are becoming more popular leading to the greater need to tighter security. Security in computing has and will always be a major factor in this industry and developers and researchers will continue to look for ways in which programmes can ensure data as safe as possible.

References:

[1] Elisa Bertino, Ravi Sandhu, "Database Security - Concepts, Approaches and Challenges," IEEE Transactions on Dependable and Secure Computing, Volume 2, No 1, Jan-Mar 2005.
[2] CristinaRibeiro, Gabriel David, "Database Preservation," Briefing Paper, Website, 6th June , 2012, http://www.digitalpreservationeurope.eu/publications/breifs/database_preservation_ribeiro_david.pdf.
[3] Pfeeger, "Security in Computing," Third Edition, 2004, Pearson Education.
[4] Shmueli, Erez, Vaisenberg, Ronen, Elovici, Yuval and Glezer, Chanan, Database Encryption - An Overview of Contemporary Challenges and Design Considerations, SIGMOD record, volume 38, No 3, 2009.
[5] Xiang Fu, Kai Qian, "SAFELI-SQL Injection Scanner Using Symbolic Execution," TAV-WEB, Workshop on testing, Analysis and verification of Web software, June 21, 2008, ACM 978-1-60558-052-4/08/07.
[6] 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), volume 1, No 25, pages 28-34.
[7] SushilJajodia, "Database Security and Privacy," ACM Computing Surveys, Volume 28, No 1, March 1996.
[8] Min-Shiang H, Wei-Pang Y, "Multi-level Secure Database Encryption with Subkeys," Data and Knowledge Engineering 22, 117-142, 1997.

[9] Kamp PH, "GBDE-GEOM Based Disk Encryption Source," BSDCon '03, pp. 57-68, 2003.

[10] Davida GI, Wells DL, Kam JB, "A Database Encryption System with Subkeys," ACM
Trans. Database System, 6, 312-328, 1981.

[11] Buehrer D, Chang C, "A Cryptographic Mechanism for Sharing Databases," The International Conference on Information & Systems, Hangzhou, China, pp. 1039-1045, 1991.

[12] Shmueli E, Waisenberg R, Elovici Y, Gudes E, "Designing Secure Indexes for Encrypted Databases", Proceedings of Data and Applications Security, 19th Annual IFIP WG 11.3, Working Conference, USA, 2005.

[13] Kuhn U, "Analysis of a Database and Index Encryption Scheme-Problems and Fixes," Secure Data Management, 2006.

[14] Bouganim L, Pucheral P, "Chip-secured Data Access: Confidential Data on Untrusted Servers," The 28th International Conference on Very Large Databases, Hong Kong, China, pp. 131-142, 2002.

[15] Hacigumus H, Iyer B, Li C, Mehrota S, "Executing SQL over Encrypted Data in the Database-Service-Provider model," The ACM SIGMOD 2002, Madison, WI, USA.

[16] Song DX, Wagner D, Perrig A, "Practical Techniques for Searches on Encrypted Data," The IEEE Security and Privacy Symposium, May, 2000.

[17] He J, Wang M, "Cryptography and Relational Database Management Systems," Proceedings of IEEE symposium on the International Database Engineering & Applications, Washington DC, USA, 2006.

[18] Chen G, Chen K, Dong J, "A Database Encryption Scheme for Enhanced Security and Easy Sharing," CSCWD, IEEE Proceedings, IEEE Computer Society, Los Alamitos, CA, 2006, pp. 1-6.

[19] A NET 2000 Ltd., "Data Sanitization Techniques," A White Paper (2010), Website, http://www.datamasker.com/datasanitization_whitepaper.pdf.

[20] A NET 2000 Ltd., "Data Scrambling Issues," A White Paper (2010), Website, http://www.datamasker.com/datascramblingissues.pdf.

[21] Huw Price, "A Short Guide to Scrambling, Masking and Obfuscating Data, Grid Tools," White Paper, Website, Http://grid-tools.com/Data_Masking.pdf.

[22] Ricardo Jorge Santos, Jorge Bernardino, Marco Vieira, "A Data Masking Technique for Warehouses, " ACM 978-1-4503-0627-0, 2011, page 61-69.