Security Issues Surrounding Data Manipulation in a Relational Database

Advanced Data Management (DT211), Dean Ryan, C11526797, 10/10/2015

Abstract - Businesses of today tend to rely on data storage companies to retain their data which may include details of finance, employees and clients to name a few. This can increasingly raise security threats towards their data and therefore the data's integrity can be very difficult to ensure when relating to a database. In the following paper, we centre on the different types of access control policies put into place to secure data. We discuss the use of data manipulating languages within a relational database and focus on potential security threats that can occur. We then finalise the report with techniques used against attacks on data in a relational database.

Keywords: Data, Databases, Relational Database, Data Manipulation, Security, Policies

1. INTRODUCTION

Databases are the most widely used mechanisms on which businesses depend. They are used for storing any kind of data. This data is stored in the form of well-structured tables. When referring to a relational database, data is stored in relational tables which means when new information is added, re-organisation of the table(s) is irrelevant. Companies now base their success around databases service companies which store their information. However, valuable information can be very hard to maintain and protect. The remainder of this research paper is organised as follows: Section two provides information on security in relational databases and the types of security issues one may face in regards to data integrity. A brief introduction of SQL is also explained. Section three will then provide information on the access control policies used to help secure correct user authentication. We describe models used within the different type's access policies such as the System R authorization model and the role based access control models. Finally in section four, security issues are discussed which can be found with the manipulation of data in relational databases systems. Techniques to prevent the attacks from Section four are also examined.

2. RELATIONAL DATABASE MANAGEMENT SYSTEMS

2.1 Relational Databases

Before relational databases came about, information was stored in flat databases which consisted of just one long file. We now have relational databases which store data into well-structured relational tables. Data is fitted into rows and columns within the tables. The unique thing about the relational database is that it's extendable. This means records which contain any information can be accessed immidiately to a corresponding column category. Although relational databases do contain the ease of extendability and other advantages such as the handling of multiple data fragments, it contains a number of secuirty issues which puts stored data at risk. To store information securely, the database needs to consist of three factors. The database should consist of availability. Availability means the database should be operastional at all time. It should consist of Confidentiality [1]. Confidentialty is a set of rules or a promise that limits access or places restrictions on certain types of information. Lastly, integrity assures information within a database can only be access and changed by users with permissions to do so. User permissions are very important within a database and will be discussed in further detail through out the paper.

In order to process data between relational databases and computer programs, the request information between two points has to be managed. This is done through what is known as a *database management system* [2]. The DBMS allows users to

update, retrieve, store and delete information through a language known as *SQL*. The SQL language is used to change data in an effort to make it more organised and easier to read.

3. ACCESS CONTROL POLICIES

Today's database systems are secured by what's known as Access control mechanisms. These mechanisms help ensure confidentially within database systems [2]. Polices of other sorts can be combined to create a more suitable system.

There are three types of access control policies as seen in figure 1; Discretionary, Roll based and the Mandatory policy.

3.1 Discretionary Access Control Policy

Discretionary policies as shown in figure 2, handle user's access to information within a database system depending on the users authorization permssions and identity We usually refer users of a database system to subjects. Permissions generally specify the access methods for each subject(s) in the system. Some subjects retain permission over a database and can grant authorizations on sets of information to other subjects.

Authorization administration is the function of granting and revoking authoizations [5]. The two most common types of administration are *centralized* and *ownership* administration. Centrailsed adminiztration allow some subjects with correct permissions to grant and revoke authorizations. Ownership administration on the other hand allow grant and revoke permissions on data objects entered by the owner of the object.

3.2 System R authorization model

The access permissions subjects have over tables in a system relate to the SQL statements that can be executed on tables. Access modes related to this module include operations such as select, insert, update and delete. Authorization administration of the model relies on delegation and ownership authorization. Any subjects granted permission of authorization may create new tables. When a subject creates a table, he/she has full ownership permissions of the table and can access all available modes to manipulate it. The subject can also delegate authorization permissions about the table to other subjects using the grant statement [3]. Subjects delegating access of other subjects can raise potential security issues when an operation such as revoke is used. Subjects which have been granted authorization by other subjects who are now revoked of their permissions raises questions like what privileges the remaining subject still have. *Revocation* generally considers the changes of timestamps in relation with granted authorizations. When authorizations are revoked from subjects, a recursive revocation takes place. It removes all authorizations of the created table from the subject committing the revoke. Because subjects tend to change data regularly, the recursive revoke takes place very often. However, this is overcome by using the non-cascade revoke which involves revoking authorization from a subject but the subject committing the revoke are not.

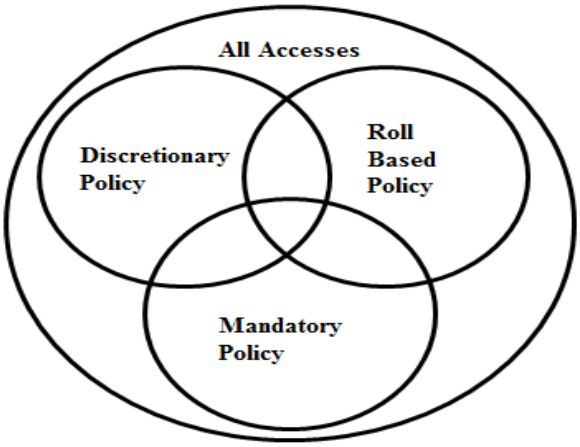

Figure 1 Different Access Control Policies [2]

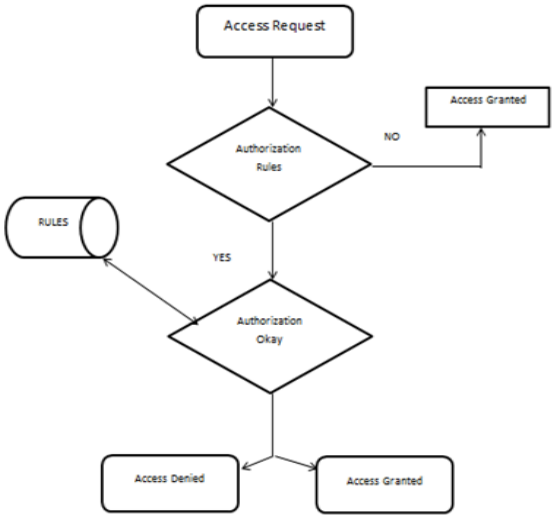

Figure 2 Discretionary Access Control Policies

3.3 Mandatory Access Control Policy

Abilties to acces or preform operations on objects can sometimes be constrained over a subject. This is whats known as *Mandatory Access Control.* Objects are usally referd to as entitites. Entities are passive in the sense that they can store a pieces of data like as the elements of a tuple or the tuples in relation. Subjects on the other hand are active in the sense that they can perform data accesses. Subjets can be granted permissions to objects if a particular order relationship relating to the access modes requirements is satisifed. The most common form of an access class would be a pair of two components. *Secuirty levels* tend to join a complete ordered set. *Categories* however are members of unordered sets [6].

Refering to the access model, access control can be based upon two main principles known as *no read-up* and *no write-down.*

No read-up: Subjects can only read dataobjects where access classes are dominated by access class of the subject. [2][6]

No write-down: Subjects may only write data objects where access classes dominate access classes of the subject. [2][6]

Furthermore, these principles refuse the flow of information within data objects through write or read operations [5].

3.4 Content Based and Fine Grained Access Control

Content based access control: The decisions of the access control in this model should be based around the contents of data.[2] The content of an object used within an application located in the content based access control model is used to tag the object with labels created by a set of features.[7]

An example of content based access control would be as follows:

Imagen a table of data stored specifically about employees of a business. The policy would state that managers are the only users that are able to access the employees who work in the project he manages.[5]

The support of content based access control within a system can be adopted into *veiws.* Views are very important in terms of support and can be split into two categories. The first being *protetction views* which help support the content of the acces control and *shorthand views* which are created to make query langauage more simplistic. [5] A view allow subjects to select subsets of columns and rows.

Fine grained access controls on the other hand supports access control at a *tuple* level. Tuple meaning an unordered set of known values. Because access control policies can be different for all users, application programs would have to be designed for each user interface which therfor means data management commands would need to use the correct view for each user or group. The *truman* model helps us deal with such issues. The truman model cancels out the individual assigining of each interface.

3.5 NIST Model

NIST is the National Institute of Standards and Technolgy to which provided technical leadership in the

stanadards and measurement infrastructure of the US. [8]

The information technology lab (ITL) carry out tests and technical analysis in order to develop an adavanced and prodictive use of information technology. The labs responsibilities include the development of technical, physical, administrative, and management standards and guidelines. [8]

3.6 *RBAC Models*

Role Based Access control bases around the concept roles. Roles can be described as particular functions used in an organisation. RBAC models are built on the purpose to try and simplify administration authorization and to represent access control policies of organisations.

All authorizations that wish to execute a given activity are assigned to the role of that activity. This cuts out direct use of the activity to the users. The process then involves assigning users as members of the roles. [5]

In addition, some RBAC contains role hierarchy. Role Hierachy allows defining of sub-role relationships. Seperation of duty constraints are also allowed which prevent users from gaining many access rights.

4. SECURITY ISSUES SURROUNDING DATA MANIPULATION

Data manipulation involves the process of deleting, inserting, modifying and retrieving information within a database. It achieves this through commands such as 'Delete', 'Update', 'Merge', 'Insert', and 'Truncate'. DML is said to be a subset of the SQL language. In order to use such language, one has to be granted the correct privileges. User privileges are assigned using the grant statement. A common grant statement might look like:] GRANT [type of permission] ON [database name]. [Table name] TO '[username]'@'. [3] Security issues do occur because of such grants and access permissions. Issues include *SQL injection*, Denial *of Service* and *Weak Authentication*.

4.1 SQL Injection

SQL injection appears when an insertion or injection is made from input by the user of an application. The injection phase can return sensitive information from the attaked database, take hold of the administartion permissions on the database, return contents of files on the database and sometimes even issue commands in the operating system.[9]

The SQL injection phase occurs when the SQL commands are injected into data-plane input. This effects the following executions of commands.

The whole idea of SQL injection is to try and convince an application to run some SQL statements that were not intended. Once the system is convinced, it will accept the statements. SQL injection contains several methods available to an attacker.

An example would be a *blind injection* attack. Attackers can use the WHERE clause which will retrieve data from the database. By adding particular commands to the SQL, the attacker could determine the vulnerabilty of an application. There are a number of reasons as to why an attacker may want to attack a system.

Identifying Injectable paramenters: involves an attack on a web application to discover if different fields and parameters are vulnearble.

*Determining a database schema:*Extracts data from the database. To do so, the attacker needs the systems schema information.

Prepared statements is one option that can be used against this attack. Also known as parameterized queries, they help force developers define the sql language.

4.2 Denial of Service

A Denial Of Service attack occurs when an attack attempts to down a system, making it inacessable to its intended users. The DOS attack is accomplished by an attacker who completely floods the target with traffic. Multiple high volume requests are sent to the server making the server overload with inforamtion forcing it to close. The attack restricts complete access to users of the service offered. There are many different types of DOS attacks which include *UDP flood*, where random ports over a network are flooded or *Ping of Deat*h which involves multiple malicious pings from an attacker to a host. An example of this attack occurred in 2013 when an activist group called LizardSquad had taken down Sony's Entertainment servers. [11]

4.3 Weak Authentication

Systems that contain weak authentication offer attackers an easy job in obtaining inforamtion. Inforamtion can include login credentials of users which allows their use of the permissions provided to the user.

Brute force is a common example of an attack which can retrieve user information. The process involves the the guessing of every possible username and password combination. A simple technique systems tend to implement to protect such attacks include strong username and password policies. Password policies suggest that the user of a system use special characters along with numbers within their passwords to prevent attacks.

5. REFLECTION

Having completed the research paper, I have learned a great deal of information about secuirty in regards to databases systems. I have learned what and why issues occur within database systems and what access control policies are and how they are used to help protect such systems. After given real world examples throughout the paper, I have learned about some of the secuirty attacks that can occur and the prevention techniques used to protect data.

6. CONCLUSION

Having completed the report, theire is no doubt the the protection of data is at severe risk in todays world. Attacks on databases are becoming more common which proves the need for secuirty techniques. User authentication as seen is very important and is expressed through the access control polcies discussed in the report. Potential secuirty attacks are explained in detail throughout the paper and example prevention techniques are provided to show how systems can protexct their information.

REFERENCES

[1] WhatIs.com, 'What is confidentiality? – Definition from WhatIs.com', 2015. [Online]. Available: http://whatis.techtarget.com/definition/confidentiality. [Accessed: 08- Nov- 2015].

[2] 'Review Of Attacks On Databases and Database Security Techniques', International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 11, pp. 1 - 11, https://www.google.ie/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCMQFjAAahUKEwjHtprfmoHJAhVHqg4KHREbATk&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.414.1729%26rep%3Drep1%26type%3Dpdf&usg=AFQjCNHcVViYAgAovTK1aZeIVMkkbp_Dbg&cad=rja , 2012.

[3] Digitalocean.com, 'How To Create a New User and Grant Permissions in MySQL | DigitalOcean', 2015. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql [Accessed: 08- Nov- 2015]

[4] 'Database Secuirty - Concepts, Approaches and Challenges', IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, vol. 2, no. 1, p. 18, 2005. https://www.google.ie/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCoQFjAAahUKEwiUyJ3jnYHJAhUEgA8KHRoTDO4&url=http%3A%2F%2Fwww.profsandhu.com%2Fjournals%2Fieee-depend-dbsec-05.pdf&usg=AFQjCNGeSInyUXz164f4MhaVRDwCgjwfkg&bvm=bv.106923889,d.ZWU&cad=rja

[5] Security, Privacy, and Trust in Modern Data Management**,** By Milan Petkovic,WillemJonker,(https://books.google.ie/books?id=Tca3vVbnRc0C&pg=PA46&lpg=PA46&dq=security+level+and+set+of+categories+access&source=bl&ots=OlQAKXzQM1&sig=8ccPZ-fXBQyrYfgMnU8wjiS9QO8&hl=en&sa=X&ved=0CCwQ6AEwAmoVChMIs4GPqbf9yAIVxmkUCh2vAw3_#v=onepage&q=security%20level%20and%20set%20of%20categories%20access&f=false)

[6] https://www.google.ie/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCkQFjAAahUKEwizsuPKwf3IAhXBcRQKHfkqB1s&url=http%3A%2F%2Fcae.ittc.ku.edu%2Fpapers%2Fcbac.pdf&usg=AFQjCNGdSW88eQN4s1KGvjXQ3E9yGHbGNw&cad=rja

[7] Peter Mell, Timothy Grance, "The NIST Definition of Cloud Computing" National Institute of Standards and Technology, Gaithersburg, MD 20899-8930 https://www.google.ie/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCgQFjAAahUKEwj6vJSon4HJAhVEHA8KHRQVCHs&url=http%3A%2F%2Fcsrc.nist.gov%2Fpublications%2Fnistpubs%2F800-145%2FSP800145.pdf&usg=AFQjCNHuHCZlkyGs_bS0ESu9p9fciwIJQA&cad=rja

[8] SQL Injection - OWASP. *[*ONLINE] Available at: https://www.owasp.org/index.php/SQL_Injection. [Accessed 11 December 2015].

[9] Sampada Gadgil, Sanoop Pillai, Sushant Poojary "SQL INJECTION ATTACKS AND PREVENTION TECHNIQUES" SIES GST Graduate School of Technology Nerul ,Navi Mumbai (http://www.academia.edu/3713854/SQL_INJECTION_ATTACKS_AND_PREVENTION_TECHNIQUES)

[11] http://krebsonsecurity.com/2014/12/cowards-attack-sony-playstation-microsoft-xbox-networks/