World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 5.218

CORE THREATS AND PREVENTION IN DATABASE SECURITY

[1]*ILO Somtoochukwu F., [2]Ubochi Chibueze and [3]Osondu U. S.

[1,2]Computer Engineering Department, Michael Okpara University of Agriculture Umudike, Abia State, Nigeria.

[3]Electronical/Electronic Engineering Department, Federal Polytechnic Nekede, P.M.B 1036, Owerri, Imo State, Nigeria.

Article Received on 10/04/2019 **Article Revised on 30/04/2019** **Article Accepted on 20/05/2019**

***Corresponding Author**
ILO Somtoochukwu F.
Computer Engineering Department, Michael Okpara University of Agriculture Umudike, Abia State, Nigeria.

ABSTRACT

Database security is a growing concern evidenced by an increase in the number of reported incidents of loss of or unauthorized exposure to sensitive data. As the amount of data collected, retained and shared electronically expands, so does the need to understand database security. Database security should provide controlled, protected access to the contents of a database as well as preserve the integrity, consistency, and overall quality of the data. At its core, database security strives to insure that only authenticated users perform authorized activities at authorized times. This paper focuses on the concepts and mechanisms particular to securing data. Within that context, database security encompasses sub-topics: these topics include Malware, Weak Authentication, Database Misconfiguration, SQL Injection, nested query and excessive privilege abuse (grant/ revoke). This work was concluded by examining data security strategies which can help us protect our system's information.

KEYWORD: Database security, database vulnerability, nested query, excessive privilege abuse, auditing mechanisms.

1. INTRODUCTION

Database security is a growing concern evidenced by an increase in the number of reported incidents of loss of or unauthorized exposure to sensitive data. As the amount of data collected, retained and shared electronically expands, so does the need to understand database

security. The main goal of database security is Information Security during all database transactions. The Defense Information Systems Agency of the US Department of Defense (2004), in its Database Security Technical Implementation Guide, states that database security should provide controlled, protected access to the contents of a database as well as preserve the integrity, consistency, and overall quality of the data therefore understanding of the issues and challenges related to database security and its possible solution should be considered.

At its core, database security strives to insure that only authenticated users perform authorized activities at authorized times. While database security incorporates a wide array of security topics, notwithstanding, physical security, network security, encryption and authentication, this paper focuses on the concepts and mechanisms particular to securing data. Within that context, database security encompasses three constructs: confidentiality or protection of data from unauthorized disclosure, integrity or prevention from unauthorized data access, and availability or the identification of and recovery from hardware and software errors or malicious activity resulting in the denial of data availability.

1.1 Database

A database is an organized collection of data (Jeffrey Ullman, 1997). The data are typically organized to model relevant aspects of reality in a way that supports processes requiring this information. Databases are created to operate large quantities of information by inputting, storing, retrieving and managing that information. Databases are set up so that one set of software programs provides all users with access to all the data.

	A	B	C	D	E
1	College Enrollment 2007 - 2008				
2					
3	Student ID	Last Name	Initial	Age	Program
4	ST348-250	Graham	J.	20	Arts
5	ST348-248	James	L.	23	Nursing
6	ST348-252	Nash	S.	22	Arts
7	ST348-249	Peterson	M.	37	Science
8	ST348-254	Robitaille	L.	19	Drafting
9	ST348-253	Russell	W.	20	Nursing
10	ST348-251	Smith	F.	26	Business
11	ST348-247	Thompson	G.	18	Business
12	ST348-245	Walton	L.	21	Drafting
13	ST348-246	Wilson	R.	19	Science
14	ST348-255	Christopher	A.	22	Science
15	*				

Figure 1: A database (Jeffrey Ullman, 1997).

1.2 Database Objects

Table: a table is the backbone of every database; it serves as a container where data are stored. A table comprises of columns (fields) and rows (records).

Field: this is a single item of information within a database. Fields run in columns, they are permanent place holders for records.

Record: this is complete information about an item. Records run in rows and keep changing

File: a file is a collection of related records within a database.

Database Management Systems (**DBMS**s) are specially designed software applications that interact with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is a software system designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, SQLite, Microsoft SQL Server, Microsoft Access, Oracle and dbase . (Beynon-Davies P. 2004).

The interactions catered for by most existing DBMSs fall into four main groups:

- **Data definition** – Defining new data structures for a database, removing data structures from the database, modifying the structure of existing data.
- **Update** – Inserting, modifying, and deleting data.
- **Retrieval** – Obtaining information either for end-user queries and reports or for processing by applications.
- **Administration** – Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information if the system fails.

1.3 Database Security

Database Security can be defined as protecting information against unauthorized disclosure, alteration or destruction using hardware or software techniques (Frank R, 1979). We can also define database security as the mechanisms that protect the databases against intentional or accidental threats. Data security includes mechanisms that control access to and use of the database at the object level, it determines which users have access to a specific schema object, and the specific types of actions allowed for each user on the object. For example, user scott can issue SELECT and INSERT statements but not DELETE statements using the employees table and define the actions, if any, that are audited for each schema object.

Data security is determined primarily by the level of security you want for the data in your database. For example, it might be acceptable to have little data security in a database when you want to allow any user to create any schema object, or grant access privileges for their objects to any other user of the system. Alternatively, it might be necessary for data security to be very controlled when you want to make a database or security administrator the only person with the privileges to create objects and grant access privileges for objects to roles and users.

Overall data security should be based on the sensitivity of data. If information is not sensitive, then the data security policy can be more lax. However, if data is sensitive, then a security policy should be developed to maintain tight control over access to objects.

Some means of implementing data security include system and object privileges, and through roles. A role is a set of privileges grouped together that can be granted to users. Views can also implement data security because their definition can restrict access to table data. They can exclude columns containing sensitive data.

2. Review of Related Article on Threat Analysis

According to literature, in 2012 Saurabh Kulkarni and Siddhaling Urolagin concluded that a database is the backbone for any type of application. The content of a database is salient. It contains very important and confidential information, so there is a chance of attacks. Various database vulnerabilities are discussed in this paper. Review of some important control methods for database threats like Discretionary Access Control Policy, Mandatory Access Control Policy, Role Based Access Control Policy and Auditing Mechanism are discussed following the study called out by Bertino et. al, he explained the discretionary access control and mandatory access control mechanisms in RDBMS. They illustrated the limitations of traditional discretionary access control and the need for mandatory access controls. In 2014, Shelly Rohilla, Pradeep Kumar Mittal explained that Databases are a favorite target for attackers because of their confidential and important data. There are many ways in which a database can be compromised. This is because there are various types of attacks and threats from which a database should be protected. According to Shivnandan Singh et al (2014) they are the primary form of storage for many organizations. So the attacks on databases are constantly increasing by the day and they are very dangerous form of attack. They reveal salient data to the attacker. Everyday critical vulnerabilities are reported on a wide variety of sensitive information in a database, and malicious activities perpetrated against it are quickly

becoming the number one security problem, which ranges between large scale social engineering attacks and exploiting critical vulnerabilities. Recent sophisticated attacks use SQL injection, nested query, excessive privilege abuse, weak authentication database misconfiguration and malware. The Evolution of new malicious code threats as reported by Symantec is shown in fiqure2.

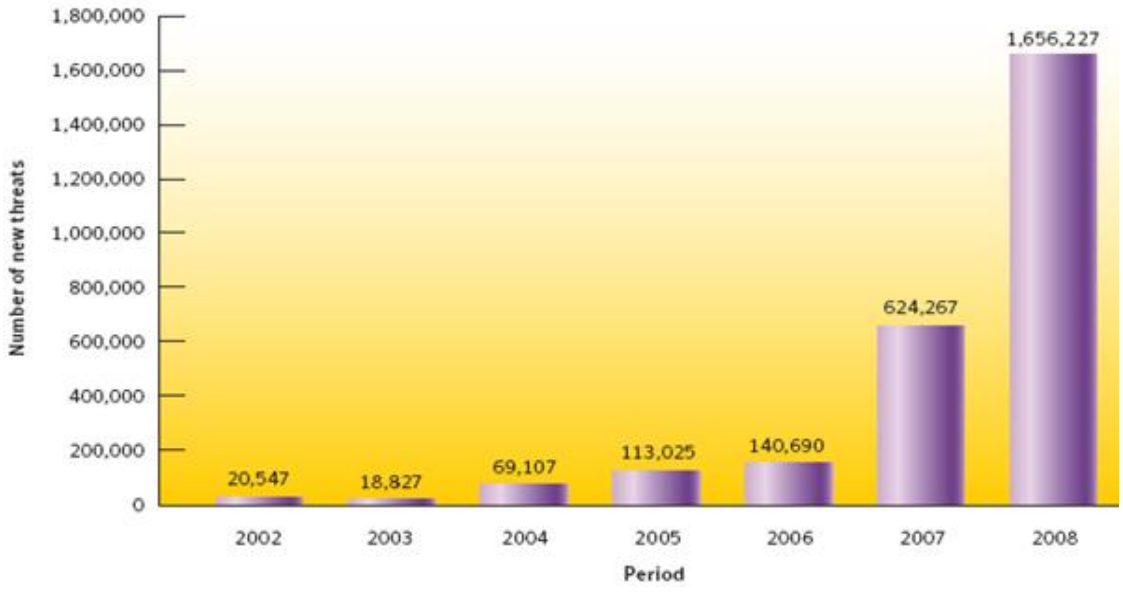

Figure 2: Evolution of new malicious code threats as reported by Symantec, (Mihai C. et al, 2008).

In figure2 reported malicious code threats detected in 2008 (1,656,227) by Symantec representing over 60 percent of the approximately total (2.6 million) malicious code threats detected in total, over time. This reality is enough to saliently reduce the validity of database content. The increase of complexity and sophistication of such attacks, the professionalization of attackers and evolution of attack patterns represent the major changes in the current threat landscape, and justify the need for urgent resolution against database insecurity. Nowadays a fast and reliable mechanism to mitigate, discern and generate vaccines for such attacks is vital for the successful protection of database contents. This research will lead to more concrete solution for database security issue.

3. Database Vulnerabilities

Today, databases are obviously faced with different kinds of attacks. The knowledge of an attack is the first step to its rectification. At the apex, the sensitivity of data determines “the if” and “the how” of data security. Describing the attacks which can be performed on the

database is very salient before describing the technique to secure the database. These attacks are further elaborated in the following sections

a. Malware - One of the most dangerous database threats we are facing today is the unprecedented spread of malware. According to Honig, A. et al, These threats can be delivered in many different variant modes often called blended threats which contain multiple components such as fishing attempts, spams, viruses, worms and Trojan. This blended threats or malware categories in percentage are shown in **Figure 3.**

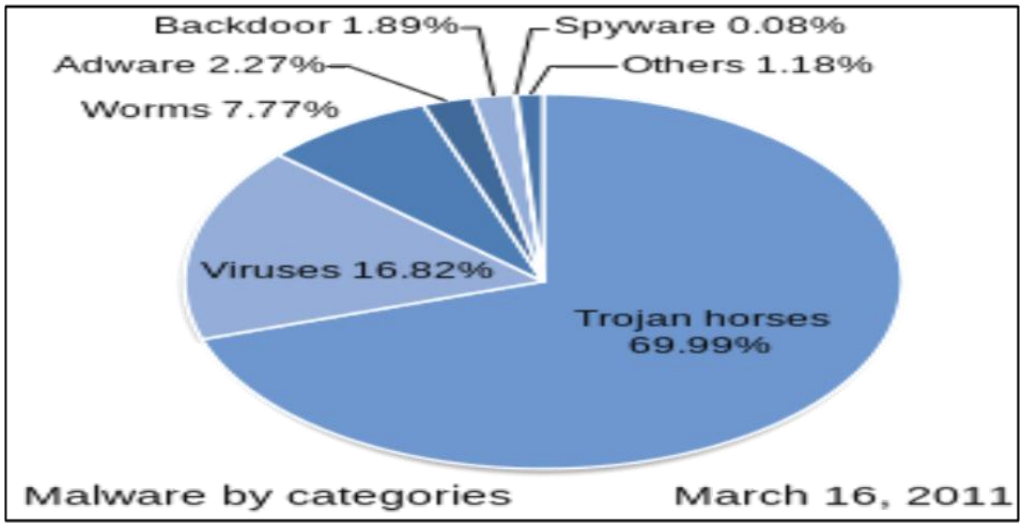

Figure 3: A Pie Chart Showing in Percentage, the Different Categories of Malware (Guri, M. et al).

According to Mubina Malik (2016), Cybercriminals, state-sponsored hackers, and spies use advanced attacks that blend multiple tactics – such as spear phishing emails and malware – to penetrate organizations and steal sensitive data. Unaware that malware has infected their device; legitimate users become a conduit for these groups to access your networks and sensitive data. Basically, to prevent malware, enable firewall protection and install Antivirus.

b. Weak Authentication - Before a data object can be accessed, the access control mechanism checks the rights of the user against a set of fixed authorizations. If the authentication mechanism is weak, it allows attackers to take up the identity of a legitimate database user. Basically, when the strength of the authentication mechanism is relatively vulnerable compared to the value of the assets being protected, then the content of the database is seriously prone to attack. Brute force, social engineering are some specific attack strategies used to bridge database security. Implementation of passwords or two-factor

authentication is a must. For scalability and ease-of-use, authentication mechanisms should be integrated with enterprise directory/user management infrastructures.

c. Database Misconfiguration - It is still very common to find un-patched databases, or randomly discover databases that have default configuration parameters. Rational attackers very well know how to exploit this ignorance and launch attack against an organization. The sad news is that organizations often struggle to stay on top of maintaining database configurations even when patches are available. Basically these scenarios include the challenge of finding a maintenance window to take down and work on what is often classified as a business-critical system, time-consuming requirements for testing patches and mounting backlogs and workloads for the associated database administrators. In the research carried out by (Mubina Malik et al, 2016), the net result is that it generally takes organizations months to patch databases, during which time they remain vulnerable.

d. SQL Injection - Security breaches are an increasing phenomenon. As more and more databases are made accessible via the Internet and web-based applications, their exposure to security threats will rise. The objective is to reduce susceptibility to these threats. Perhaps the most publicized database application vulnerability has been the SQL injection. SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed. It is an instance of a more general class of vulnerabilities that can occur whenever one programming or scripting language is embedded inside another. SQL Injection is one of the most common application layer attack techniques used today. SQL injection attacks are also known as SQL insertion attacks.

e. Rephrased, this means using special input to trick the SQL server. SQL injections provide excellent examples for discussing security as they embody one of the most important database security issues, risks inherent to non-validated user input. SQL injections can happen when SQL statements are dynamically created using user input. The threat occurs when users enter malicious code that ‘tricks’ the database into executing unintended commands. The vulnerability occurs primarily because of the features of the SQL language that allow such things as embedding comments using double hyphens (--), concatenating SQL statements separated by semicolons, and the ability to query metadata from database data dictionaries.

A common example depicts what might occur when a login process is employed on a web page that validates a username and password against data retained in a relational database. The web page provides input forms for user entry of text data. The user-supplied text is used to dynamically create a SQL statement to search the database for matching records. The intention is that valid username and password combinations would be authenticated and the user permitted access to the system. Invalid username and passwords would not be authenticated. However, if a disingenuous user enters malicious text, they could, in essence, gain access to data to which they have no privilege. For instance, the following string, ' OR 1=1 --entered into the username textbox gains access to the system without having to know either a valid username or password. This hack works because the application generates a dynamic query that is formed by concatenating fixed strings with the values entered by the user. For example, the model SQL code might be:

SELECT Count (*) FROM UsersTable
WHERE UserName = 'contents of username textbox'
AND Password = 'contents of password textbox';

When a user enters a valid username, such as 'Mary' and a password of 'qwerty', the SQL query becomes:

SELECT Count (*) FROM UsersTable
WHERE UserName='Mary'
AND Password='qwerty';

However, if a user enters the following as a username: 'OR 1=1 -- the SQL query becomes:

SELECT Count (*) FROM UsersTable
WHERE UserName=''OR 1=1 - -'
AND Password='';

The expression 1 = 1 is true for every row in the table causing the OR clause to return a value of true. The double hyphens comment out the rest of the SQL query string. This query will return a count greater than zero, assuming there is at least one row in the users table, resulting in what appears to be a successful login. In fact, it is not. Access to the system was successful without a user having to know either a username or password.

f. Nested Query - Another SQL injection is made possible when a database system allows for the processing of stacked queries. Stacked queries are the execution of more than one SQL query in a single function call from an application program. In this case, one string is

passed to the database system with multiple queries, each separated by a semicolon. The following example demonstrates a stacked query. The original intent is to allow the user to select attributes of products retained in a Products table. The user then injects a stacked query incorporating an additional SQL query that also deletes the Customers table, going beyond the intended privilege given to user.

Select * From Products; Drop Customers

This string when passed as an SQL query will result in the execution of two queries. A listing of all information for all products will be returned. In addition the Customers table will be removed from the database. The table structure will be deleted and all customer data will be lost. In database systems that do not allow stacked queries, or invalidate SQL strings containing a semicolon, this query would not be executed.

g. Excessive Privilege Abuse (Grant/ Revoke) - When users (or applications) are granted database access privileges that exceed the requirements of their job function, these privileges may be abused for malicious purpose. 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 grades.

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.

4. Database Security Control

The following gives us different ways core database security threats are being controlled.

1. Access Control Policies (Authentication and Authorization) - Access Control Mechanisms are used for securing databases. It ensures data confidentiality. Whenever a user tries to access a data object, the access control mechanism checks the rights of the user against a set of fixed authorizations. Different policies can be combined to provide a more suitable protection to database system. There are two main access control policies - Mandatory Access Control Policy and Discretionary Access Control Policy. In modern age new access control policy -Role Base Access Control is used. The RBCA is most popular access control model used to secure data in a database.

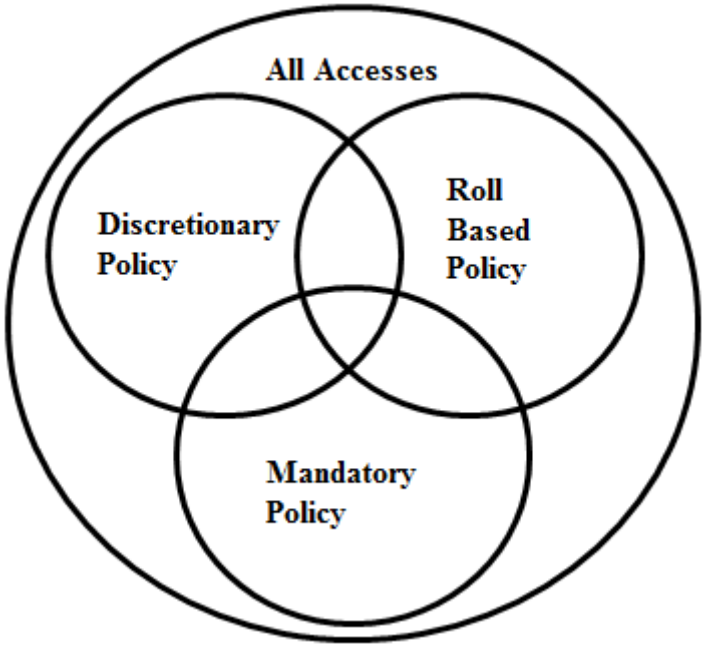

Figure 4: Different Access Control Policies (Ravi S. 1996).

a. Discretionary Access Control Policy - Discretionary protection policies govern the access of users to the information on the basis of the user's identity and authorizations. These authorizations are also known as rules. These rules specify the access modes, for each user (or group of users) and each object in the system. Discretionary Access Control (DAC) can be referred as a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. This policy places the decision of who can access information at the discretion of the information creator i.e. owner of data or database administrator. Security policy implementation is based on granting and revoking privileges. Access is granted or denied based on the identification of the user. The Authorization Administration Policy supervises this function in DAC. Common Administration Policies used in DAC are Centralized Administration and Ownership Administration. In centralized administration only some privileged subjects may grant and revoke authorizations while in ownership administration grant and revoke operations on data objects are entered by the creator (or owner) of the object (Ravi S. 1994). User-level privileges in DAC define access permissions based on the general account information of user, whereas the relation-level privileges control access to the individual relations of the database (Feikis John 1999). The figure below gives basic implementation of DAC. The flexibility of discretionary policies makes them suitable for a variety of systems and applications.

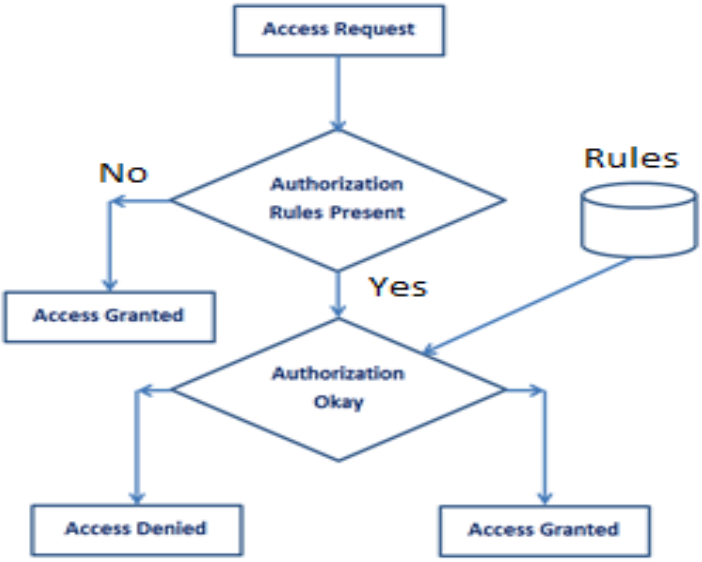

Figure 5: Discretionary Access Control Policy (Feikis John 1999).

b. Mandatory Access Control Policy - Mandatory Access Control (MAC) constrains the ability of a user to access or generally perform some sort of operation on database objects. MAC policy requires all users to follow the rules of access set up by the Database Administrator (DBA). This policy needs objects (e.g. Database) to be classified and subjects (e.g. Users, Process) to be cleared. It is enforced by comparing attributes of a subject and an object to control access to the object. It restricts access to objects based on the sensitivity of the information. (Feikis John, 1999).

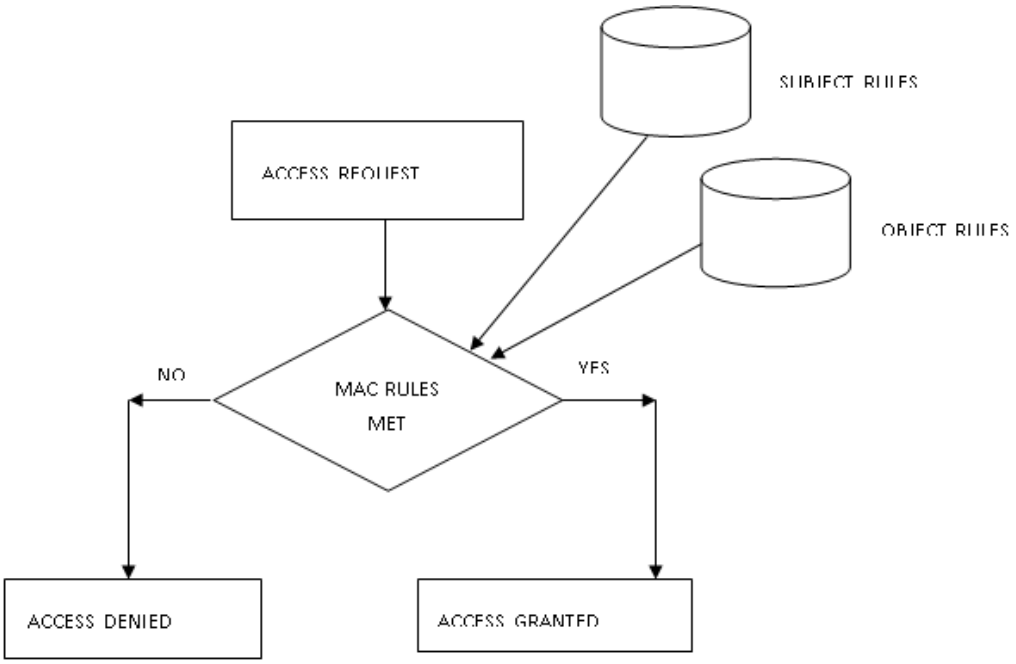

Figure 6: Mandatory Access Control Policy (Feikis John 1999).

c. Role Based Access Control Policy - Role Based Access Control (RBAC) models represent the most important recent innovation in access control model. RBAC has been motivated by the need to simplify authorization administration and to directly represent access control policies of organizations. Role-based policies regulate users' access to the information on the basis of the activities the users execute in the system i.e. RBAC models are based on the notion of role. A Role represents a specific function within an organization and can be seen as a set of actions or responsibilities associated with this function. Under an RBAC model, all authorizations needed to perform a certain activity are granted to the role associated with that activity, rather than being granted directly to users. Users are then made members of roles, thereby acquiring the roles' authorizations. Thus user access to objects is mediated by roles; each user is authorized to play certain roles and, on the basis of these roles, a user can perform accesses to the objects. (Bertino Elisa, 2005)

2. Preventing SQL Injection

a. Input validation - One effective prevention against SQL injection attacks is due diligence on the part of the web programmer. The programmer must insure that all text entered into HTML input fields is validated and stripped of malicious SQL content before it is sent to the database. This is an example of the "Secure By Design" philosophy. NEVER take user input and place it directly into a SQL query. Always sanitize user input. Watch for characters like ',", _,%,\x00,\n,\r,\, and \x1a. If possible create a white list of what characters are acceptable, and don't make it contain any more than you need.

b. Minimum Password Length - The MinPasswordLen property sets the minimum length of a user password. If set to 0 then the minimum password length is not limited. By setting the minimum password length you can prevent using too short passwords, therefore lowering the possibility of a password to be guessed by an intruder. This property is set by the security administrator.

c. Establish strong password: Implementing strong passwords is the easiest thing you can do to strengthen your security against SQL Injection. Use a combination of capital and lower-case letters, numbers and symbols and make it 8 to 12 characters long. Avoid using: any personal data (such as your birthdate), common words spelled backwards and sequences of characters or numbers, or those that are close together on the keyboard.

3. Auditing Mechanism

Just as video cameras supplement audible alarms on doors and windows in homes and businesses, auditing provide the corresponding who, what, and when that complement database monitoring.

Database auditing is used to track database access and user activity. Auditing can be used to identify who accessed database objects, what actions were performed, and what data was changed. Database auditing does not prevent security breaches, but it does provide a way to identify if breaches have occurred. Common categories of database auditing include; monitoring database access attempts, Data Control Language (DCL) activities, Data Definition Language (DDL) activities, and Data Manipulation Language (DML) activities (Yang, 2009). Monitoring database access attempts includes retaining information on successful and unsuccessful logon and logoff attempts. DCL audits record changes to user and role privileges, user additions, and user deletions. DDL audits record changes to the database schema such as changes to table structure or attribute data types. DML audits record changes to data. Database auditing is implemented via audit tables.

Auditing is generally used to:

- Enable future accountability for current actions taken in a particular schema, table, or row, or affecting specific content
- Investigate suspicious activity. For example, if an unauthorized user is deleting data from tables, then the security administrator could audit all connections to the database and all successful and unsuccessful deletions of rows from all tables in the database.
- Monitor and gather data about specific database activities. For example, the database administrator can gather statistics about which tables are being updated, or how many concurrent users connect at peak times.

Audit trail records can contain different types of information, depending on the auditing options set. The following information is always included in each audit trail record, if the information is meaningful to the particular audit action:

- User name
- Password
- Name of the schema object accessed
- Operation performed or attempted
- Date and time

- System privileges used (Yang, 2009)

3.1 Basic Architecture Of Auditing System - The basic architecture of an auditing system is:

Figure 7: Anatomy of Auditing System (Bishop 2003).

The logger records information. The information that is stored in the log is determined by the security and the system capabilities. The analyzer takes the log as the input and analyzes it to either determine if an event of interest or a problem has occurred, or if other information needs to be logged. The notifier receives the analysis from the analyzer, and reports the result of the audit to the analyst, auditor, or other entities such as GUI form. In the following sections we will describe features of each component in detail and provide a framework to classify database auditing systems.

a. Logger - Some important issues related to logging include what, when, where, how, and how often a database was logged. More customized auditing procedures can be written for table-oriented logging systems. The mechanisms used to set the type of event and condition to be monitored address the "how" question of logging. A variety of sanitization and encryption techniques have been utilized by database auditing systems. The extent of auditing required and/or desired is the most essential factor in determining what to log by the Logger Analyzer Notifier Anatomy of Auditing System (Bishop 2003) because the ability to reconstruct an event which is of interest to the analyst depends on how accurately a database has logged the data for those suspicious events. An entry in a log must contain sufficient information to find the consequence of a certain action. Nonetheless, monitoring for critical tables and system is required.

b. Analyzer - In analyzing there are two major issues: what to analyze for and when to analyze. The question, "what to analyze" considers the goals of auditing, which are usually related to the detection of security violations. Regarding "when to analyze", the analysis can be periodic, based on transaction counting, and/or occur in real-time.

There are two major categories of frequency of analysis: periodic and real-time (Orman 1997). More frequent analysis can detect violations faster and allow prompt settlement. Real-

time analyses analyze and validate each transaction and report immediately of the policy violation. Real-time analysis provides most frequent analysis and is done for policies that are critical to the system. Periodic analyses perform analysis after a given period of time. This method is most often employed to give insight of the overall system transitions via providing periodic reporting.

c. Notifier - There are two issues in the notification stage that are of primary interest. The first is the audit browsing techniques available. The second is the ability of the notification system to sanitize the data based on the policy and the viewer's privilege level. The goal of the audit browsing tools is to be able to present information in a way that it is easy for the security administrators and auditors to identify potential threats or violations of policy. Before a user is allowed to view the log or result of analysis, information that is not relevant or information level that is higher than user's allowed access, is removed. In the context of database auditing, it is salient that users be restricted to search and view entries of a narrow scope so no user is allowed to access all entries.

4. CONCLUSION

The need to secure computer systems is well understood and securing data must be part of an overall computer security plan. Growing amounts of sensitive data are being retained in databases and more of these databases are being made accessible via the Internet. As more data is made available electronically, it can be assumed that threats and vulnerabilities to the integrity of that data will increase as well. Failing to safeguard databases that store sensitive data can cripple your operations and result in regulatory violations. Understanding database threats and implementing the solutions outlined will enable you to recognize when you're vulnerable or being attacked, maintain security best practices and ensure that your most valuable assets are protected.

REFERENCE

1. Abel D. J., Relational Data Management Facilities for Spatial Information Systems. Proceedings of the 3rd International Symposium on Spatial Data Handling International Geographical Union. Columbus Ohio, 1998; 9-18.
2. Abel D. J., Smith J. L., A Relational Database Accommodating Independent Partitions of the Region. Proceedings of the 2nd International Symposium on Spatial Data Handling. International Geographical Union. Columbus Ohio, 1986; 23-24.

3. Anil L. Pereira, Vineela Muppavarapu and Soon M. Chung Role-Based Access Control for Grid Database Services Using the Community Authorization Service‖, IEEE Transactions on Dependable and Secure Computing, 2006; 3(2).
4. Bertino, E., & Sandhu, R. Database Security—Concepts, Approaches, and Challenges. IEEE Transactions on Dependable and Secure Computing, 2005; 2(1): 2-18.
5. Bertino, E., Byun, J., & Kamra, A. Database security. In M. Petkovic & W. Jonker (Eds.), security, privacy, and trust in modern data management (Data-centric systems and applications), 2007; 87-102.
6. Beynon-Davies P. Database Systems 3rd Edition. Palgrave, Basingstoke, UK, 2004.
7. Bishop A. Anatomy of Auditing System, 2003.
8. Chen P. The Entity-Relational Model – Towards A Unified View Of Data. Association for computing machinery transaction on database systems, 1976; 9-36.
9. Clark D. M., Hastings D A, Kineman J. J. Global Database and Their Implications, 1991.
10. Codd E. F. Extending the Database Relational Model To Capture More Meaning. Association for Computing Machinery Transactions on Database Systems, 1979; 397-434.
11. Codd E. F., A Relational Model of Data for Large Shared Data Banks. Communications of the Association for computing machinery, 1970; 377 – 87.
12. Collins J. Review of Competitive Database Software, 1982.
13. Fagin R. Normal Forms and Relational Database Operations, 1979; 153-60.
14. Feikis John Database Security: IEEE Journals, 1999.
15. Frank A U. Requirements for A Database Management System Based On Linear Quadrees And A Relational Database For Regional Analysis, 1988; 213-32.
16. Frank R. Requirements for Database Systems Suitable To Manage Large Spatial Database. Proceedings of The 1st International Symposium On Spatial Data Handling, 1979; 1: 38-60.
17. Guimaraes, M. New challenges in teaching database security. Proceedings of the 3rd Annual Conference on Information Security Curriculum Development, Kennesaw, GA, USA, 2006; 64-67.
18. Guri, M. et al, Y. "BitWhisper: Covert Signaling Channel between Air-Gapped Computers Using Thermal, 2015.
19. Honig, A. et al. Practical Malware Analysis. No Starch Press. Retrieved 5 July, 2012.
20. International Journal of Computer Science and Information Technologies, 2014; 5(3).

21. Jaquith, A. Security metrics: Replacing fear, uncertainty, and doubt. Redwood City, CA: Addison Wesley Professional, 2007.
22. Jeffrey Ullman First course in database systems, Prentice–Hall Inc., Simon & Schuster, 1997; 1.
23. Journal of Advanced Research in Computer Science and Software Engineering, 3(5).
24. Knox, D. C. Effective Oracle database 10g security by design. New York: McGraw-Hill/Osborne, 2004.
25. Marius ConstantinLeahu, Marc Dobson, and Giuseppe Avolio. Access Control Design and Implementations in the ATLAS Experiment‖, IEEE Transactions on Nuclear Science, May 2013; 55(1).
26. Mihai Christodorescu, et al. Malware threats and mitigation strategies: US-CERT informational whitepaper, 2013.
27. Mr. Saurabh Kulkarni, Dr. Siddhaling Urolagin, Review of Attacks on Databases and Database.
28. Mubina Malik database security – attacks ans control methods. International Journal of Information Sciences and Techniques (IJIST), 2016; 6(1/2).
29. Orman A. Database Security, 1997.
30. Ravi S. Sandhu and Pierangela Samarati, Access Controls Principles and Practice‖, lEEE Communication\ Magazine, 1994.
31. Ravi S. Sandhu, Edward J. Cope, Hal L. Feinstein, Charles E. Youman. Roll Based Access Control Models‖, IEEE Journals, 1996.
32. Security Techniques, Facility International Journal of Engineering Technology and Database Security.
33. Shelly Rohilla, Pradeep Kumar Mittal, Database Security: Threats and Challenges, International.
34. Shivnandan Singh, Rakesh Kumar Rai, A Review Report on Security Threats on Database.
35. Staddon, J. Dynamic inference control. Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, CA, USA, 2003; 94-100.
36. Techniques Research, November-2012; 2(11).
37. The Defense Information Systems Agency of the US Department of Defense, 2004.
38. Yang, L. Teaching database security and auditing. Proceedings of the 40th ACM Technical Symposium on Computer Science Education, Chattanooga, TN, USA, 2009.