Open Access Library Journal
2024, Volume 11, e11366
ISSN Online: 2333-9721
ISSN Print: 2333-9705

Analysis of Database Security

Xueying Pan, Anthony Obahiaghon, Brendan Makar, Samuel Wilson, Christian Beard

Department of Computer Science & Engineering, Oakland University, Rochester, USA
Email: ivypan89@gmail.com

How to cite this paper: Pan, X.Y., Obahiaghon, A., Makar, B., Wilson, S. and Beard, C. (2024) Analysis of Database Security. *Open Access Library Journal*, **11**: e11366.
http://doi.org/10.4236/oalib.1111366

Received: February 27, 2024
Accepted: April 12, 2024
Published: April 15, 2024

Abstract

The purpose of this paper is to analyze a variety of factors arising from database vulnerabilities such as software bugs, misconfigurations, insecure coding practices, and security threats, and to discuss how database administrators (DBAs) response to these database vulnerabilities and threats. In this paper, we not only discuss how authorized users use various techniques to secure data schemes, get privileged access, and keep database system security but also introduce different control measures and mechanisms for granting and revoking privileges in the relational database system. Specifying security mechanisms including discretionary access control, mandatory access control, role-based access control, and Extensible Markup Language (XML) access control against different database threats such as Structured Query Language (SQL) injection attacks that would have caused loss of integrity, availability, and confidentiality. We have addressed specific preventive measures to the one of major database threats, which is SQL injection. From deeply analyzing statistical database security, we have found security problems that need us to pay attention to flow control and covert channels. Finally, we summarized some of the key research results including vulnerability analysis, threat modeling, access control mechanisms, cryptographic techniques, and database forensics.

Subject Areas

Database Security

Keywords

Database Security, SQL Injection, Control Measure, Security Mechanism, Key Infrastructure, Database System

1. Introduction

As database management is growing day by day, most of the people are not aware of security and privacy. The database system is widespread information infrastructure. It is an insecure database management for end users to access the database system. The security of the database system is the set of rules and control measures taken against various database threats. Most of websites and applications have security vulnerabilities that make threats possible. Some attacks may be possible due to loss of integrity, availability, confidentiality. The others may occur by SQL Injection when having a poorly written code of web application or configuration mistakes existed. All improper modifications in the database system may cause integrity to be lost which makes end users use incorrectly modified data of the contaminated system to have inaccuracy or fraud decisions. If a human user or a program had an illegal right to data objects which would have resulted in a loss of availability that means those who cannot access these objects. All unauthorized disclosures have a bad impact on database confidentiality since they not only violate the Data Privacy Act but also jeopardize national security. It is, therefore, of the huge importance of preventing such types of security threats to databases, and SQL injection prevention methods have become one of the most common topics of research in the governmental, institutional, and corporate fields.

For former researchers, they may encounter some common challenges and situations when working with sensitive data that raises concerns in data privacy and compliance with regulations such as PCI DSS, HIPAA, and GDPR. Data anonymization and regulatory compliance would be difficult in conducting security analysis. Complex and diverse database systems have various components, configuration, and deployment models that contribute to the complexity of database architectures, query languages, interaction patterns, and data models then former researchers have to face these challenges. They may struggle to keep eye on the latest trends and threats in analyzing database security because new attacks techniques, vulnerabilities, and exploits emerging regularly require them to continue learning and adapting to threat landscape in database security. To conduct useful security analysis of database, former researchers may have challenges to get access to diverse real-world data sets and environments for further testing, deep evaluation, and validation of techniques in security analysis. Limited funding and infrastructure in academic and nonprofit settings would be an issue when they want to conduct comprehensive security analysis within secured the necessary resources because requirements of significant resources cannot reach in time, expertise, computational power, and financial investment.

In this paper, we have caught up with specific preventive measures for database threats. Analyzing security problems in statistical databases needs people to focus on studying flow control and covert channels. Keeping data encryption based on symmetric key and asymmetric key infrastructure schemes and digital certificate design. Specific analysis of Privacy-preserving techniques ensures the privacy of data and improves data preservation. Discussion of current challenges in maintaining database security would make people beware of further studying needed in data quality, integrity checking semantics, IPR, and database survivability. Industries and business organizations have commonly used Oracle label-based security to handle their sensitive and classified data including government agencies, healthcare organizations, financial institutions, and defense contractors. This paper explained how Oracle label security would be leveraged for business organizations to enforce policies of data security, prevent unauthorized access to confidentially sensitive information, and reach regulatory requirements in maintaining compliance.

2. Control Measures to Security of Data in Database

To enforce database system security, we provide four main types of control measures including access control, inference control, flow control, and data encryption. For controlling the login process, DBMS uses access control to create user accounts and passwords. As a result, it could prevent unauthorized persons from accessing the database systems to obtain confidential information or making malicious changes in a portion of the database. Especially, in the statistical database system, security methods focus on keeping individual information not being accessed from database queries. To protect the statistical database, there is the second control measure, called inference control. When unauthorized users want to access detailed confidential information by flowing in such a way, we could use a flow control measure to stop them. Moreover, we could know which information flows implicitly in ways and violate the security policy by analyzing convert channels even if there still have some issues in covert channels. Finally, data encryption would be the fourth control measure in order for protecting sensitive data such as credit card numbers, social security numbers because both of them belong to personal information that could use some coding algorithm methods to make unauthorized users have difficulty deciphering encoded data. This also makes those who have difficulty decoding encryption techniques without decrypting keys. In the other hand, developing encrypting techniques have used in military applications. For private organizations and governmental applications, they are both using encrypted database records. To support web-based transactions again database shortcomings, public key encryption would be an encryption technique. For securing personal communications, we would use digital signatures as another encryption technique.

3. Database Security and the DBA

To manage a database system, database administrators have to take actions including account creation, privilege granting, privilege revocation, and security level assignment to control a group of users who need to access DBMS with certain privileges. If some privileges previously have been given to specific accounts, the database administrators could be able to revoke or cancel certain privileges. For every user account, it needs to be assigned to the appropriate security clearance level in accordance with the policy of the organization. The main purpose of granting certain privileges, revoking privileges, and as-signing security level, is to control discretionary database authorization and to control mandatory authorization.

For database administrators (DBAs), they have faced kind of challenges related to database security. (See Figure 1) First of all, there are some of external threats from hackers and internal threats from employees or end-users who get unauthorized access to sensitive data with malicious intent or compromised credentials. Secondly, severe financial and reputational consequences for organizations may result from data breaches which make unauthorized disclosure of sensitive information including personal data, financial records, and intellectual property and DBAs may not ensure protected databases. The third reason of insider threats risks is from partners, employees, and contractors who may get affected by intentional or unintentional actions that compromise data security, such as negligence, data theft, or sabotage when they have legitimately accessed to database. Weaken authentication and authorization mechanisms have affected on control access to the database which some users have inappropriate permission to perform their tasks. Both of inadequate authentication and authorization could contribute to data breaches and unauthorized access.

To protect sensitive data, DBAs should run strong encryption process in case of unauthorized access, but they have come across challenges in weak encryption algorithms and improper key management in transit and at the rest periods.

4. Sensitive Data and Types of Disclosures

For the different value of data, we could separate them as sensitive data or non-sensitive data based on several important factors. For example, employees' salaries and patient's medical records are inherently sensitive data so they should be confidential. For some data, if the source of data or owner of data may have indicated they must be kept the secret, and hence they would be classified as a sensitive source and sensitive data. Such salary attributes of employees and salary history records in the personal database have been declared sensitive.

For database administrators, they are mainly responsible for enforcing security policies of a firm, which means whether end users or categories of users should be permitted to access to certain database attribute. Therefore, we should carefully think of some vital factors for making the correct decision of whether it is safe to reveal the data. To update a field, data may temporarily not be available to users, as users should not review inaccurate data. Normally, a concurrency

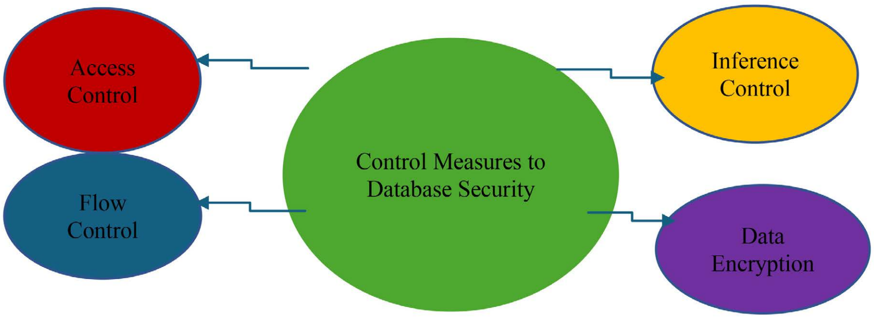


Figure 1. Main control measures for database security.

control mechanism deals with data availability issues. Of course, database administrators could be able to deny access to a user request if some data should only be available to authorized users. This is, called access acceptability, as the second factor of revealing data. From authenticity assurance, it may make users only have permission to access certain databases during working hours in order to ensure a combination of queries does not reveal sensitive data.

5. Relationship between Information Security and Information Privacy

To answer a question to the relationship between information security and information privacy, we have to do some research in below five aspects to protect the database system. Firstly, authentication of users must be strong because weak authentication would give attackers opportunities to get legitimate rights from targeted users and then steal or modify credentials in database systems. A social engineer is a common example when hackers want to get password information by using phone calls. Secondly, information encryption always becomes an effective method to make data not be read and understood easily by unauthorized users since it uses a special algorithm to encode database system and only could be accessible when decryption key is given. Thirdly, access control is to control users who want to access to database systems when they provided correct usernames and passwords. Fourthly, firewall policy should be set based on the database control mechanism in order to prevent people from getting unauthorized access to the database systems on restricted areas. Finally, intrusion detection primarily monitors database systems for malicious activity or policy violations. For example, database administrators always use security information and event management (SIEM) systems to report any malicious activity or violation in that it not only combines outputs from multiple sources but also provides alarm filtering techniques to distinguish unnormal activity and malicious traffic from false alarms. From the above aspects, we could find information privacy and security which are important components of database systems.

For information privacy, we find it not only prevents the storage of personal information but also ensures the appropriate use of personal information since it purposely focuses on controlling the terms under which end users' personal information is acquired and used. What's more, it uses mechanisms to support compliance with basic principles and other stated security policies. For instance, when it comes to patients' medical information, privacy always has been discussed for protecting their medical records in the database system. However, privacy threats may arise because users could have used data mining tools or come from information retrieval to get information which would cause the information to be used in a wrong or incorrect manner. Therefore, we should carefully release information to make sure privacy violations would not be triggered.

For information security, it mainly involves a kind of technology in order to make sure related information has been appropriately protected. At the same time, we should pay more attention to three issues including authenticity, integrity, and confidentiality when dealing with information security issues. The main reasons are to make sure received information is coming from the source from which we claim, and ensuring information is not modified during its transferring periods from the source to the targeted recipients. Finally, we need to keep the information confidential, which means data only could be reviewed by authorized requestors based on disclosure and access control policies. GDPR, HIPAA, SOX, and PCI DSS are regulatory requirements and industry standards which ensure database comply with these compliance requirements and improve data security from implementing appropriate security controls and auditing mechanisms in business organizations.

6. Discretionary Access Control Based on Granting and Revoking Privileges

In a DBMS, granting and revoking privileges are the typical method of enforcing discretionary access control. There are two types: account level and relation (or table) level. At the account level, privileges that each account holds independent of the relation in the database are specified by the DBA, while at the relation level, the privilege to access each individual relation or view in the database is controlled by the DBA [1].

7. Specifying Privileges through the Use of Views

Views are an important discretionary authorization mechanism. It is created by defining the view by means of a query that selects only those tuples from R that A wants to allow B to access.

8. Revoking of Privileges

When temporarily granting a privilege to a user for a specific task, there will need to revoke the privilege at the end of the task. The REVOKE command is included in SQL for the purpose of canceling privileges.

9. Propagation of Privileges Using the Grant Option

When the owner A of a relation R grants a privilege on R to another account B, the privilege can be given to B with the GRANT OPTION. That means B can also grant that privilege on R to other accounts. By so doing, privilege on R can propagate to other accounts without the knowledge of the owner R. If the Owner account A decides to revoke the privilege granted to B, all the privileges that B propagated based on that privilege will automatically be revoked by the system.

10. Specifying Limits on Propagation of Privileges

Different techniques to limit the propagation of privileges have been developed, but not been implemented yet. Limiting horizontal propagation to an integer number I means that an account B given the GRANT OPTION can grant the privilege to at the most I other accounts. While vertical propagation limits the depth of the granting of privileges, granting a privilege with vertical propagation of zero is equivalent to granting the privilege with no GRANT OPTION.

11. Mandatory Access Control and Role-Based Access Control for Multilevel Security

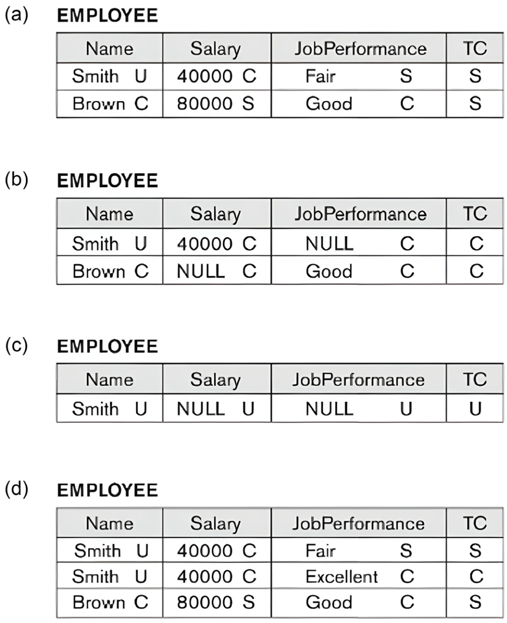
Although most mainstream RDBMS provide mechanisms just for discretionary access control, there is still the need for multilevel security because of the importance of information privacy. There are different security classes: top secret (TS), secret(S), confidential (C), and unclassified (U), where U is the lowest level and TS is the highest level. The Bell-LaPadula model is commonly used for multilevel security, which classifies each subject and object into one of the security classification TS, S, C, or U. The clearance of a subject S is referred to as class (S) and the classification of an object O as class (O). There are two restrictions that are enforced on data access based on the subject/object classification. The first states that no subject can read an object whose security classification is higher than the subject's security clearance. The second restriction prohibits a subject from writing an object at a lower security classification than the subject's security clearance. When these rules are violated information flows from a higher classification to lower classification [1].

The classification at multiple security levels is known as the multilevel model which describes each attribute A is associated with a classification attribute C in the schema, and each attribute value in a tuple is associated with a corresponding security classification. Also, a tuple classification attribute TC is added to the relation attributes to provide a classification for each tuple as a whole. The apparent key of a multilevel relation is a set of attributes that would have formed the primary key in a regular (single-level) relation. Filtering is the process of storing a single tuple in the relation at a higher classification level and produce the corresponding tuple at the lower-level classification. The concept where severe tuples can have the same apparent key value but have different attribute values for users at different clearance levels is called polyinstantiation.

The above **Figure 2** is a multilevel relation to illustrate multilevel security. (a) The Original EMPLOYEE tuples, (b) Appearance of EMPLOYEE after filtering for Classification C users, (c) Appearance of EMPLOYEE after filtering for classification U users, (d) Polyinstantiation of the Smith tuple.

12. Comparing Discretionary Access Control and Mandatory Access Control

In comparing discretionary access control (DAC) and Mandatory access control (MAC) we need to look at their advantages and disadvantages. DAC is suitable for a large variety of application domains because its policies are characterized by a high degree of flexibility. But its disadvantage is that it is vulnerable to malicious attacks because it does not impose any control over how information is



(a) **EMPLOYEE**

Name	Salary	JobPerformance	TC
Smith U	40000 C	Fair S	S
Brown C	80000 S	Good C	S

(b) **EMPLOYEE**

Name	Salary	JobPerformance	TC
Smith U	40000 C	NULL C	C
Brown C	NULL C	Good C	C

(c) **EMPLOYEE**

Name	Salary	JobPerformance	TC
Smith U	NULL U	NULL U	U

(d) **EMPLOYEE**

Name	Salary	JobPerformance	TC
Smith U	40000 C	Fair S	S
Smith U	40000 C	Excellent C	C
Brown C	80000 S	Good C	S

Figure 2. A multilevel relation to illustrate multilevel security.

propagated and used once it has been accessed by authorized users. While MAC policies ensure a high degree of protection, which prevents any illegal flow of information. Making it suitable for military and high-security types of applications, which require a higher degree of protection. The disadvantage is that it is too rigid, because they require strict classification of subject and object into security levels, therefore making it applicable to a few environments.

13. Role-Based Access Control

Many DBMS have allowed the concept of roles, where privileges can be assigned to roles. Role-based access control (RBAC) emerged in the 1990s as a proven technology for managing and enforcing security in large-scale enterprise. Security privileges that are common to a role are granted to the role name, and any individual assigned to this role would automatically have those privileges granted. RBAC can be used with traditional discretionary and mandatory access control, by ensuring that only authorized users in their specific roles are given access to certain data or resources. Separation of duties is another important requirement in various mainstream DBMS. It prevents one user from doing work that requires the involvement of two or more people, thus preventing collision.

The natural way to organize roles to reflect the organization's lines of authority and responsibility in RBAC is role hierarchy. It involves choosing the type of hierarchy and the roles and then implementing the hierarchy by granting roles to other roles [2].

14. Label Based Security and Row-Level Access Control

Most RDBMS use the row-level access control concept, in which sophisticated access control rules can be implemented by using data row by row. Each data row is given a label, which is used to store information about data sensitivity. Row-level access control allows permissions to be set for each row and not just for the table or column. Labels are used to prevent unauthorized users from viewing or altering certain data. Where a user having a low authorization level, usually represented by a low number, is denied access to data having a higher-level number. A row label is automatically assigned if no label is given to a row.

However, a label security policy is a policy defined by the administrator. When data affected by the policy is accessed or queried through an application, the policy is automatically invoked. A new column is added to each row in the schema when a policy is implemented. The added column contains the label for each row that reflects the sensitivity of row as per the policy.

15. XML Control

The increased use of XML in commercial and scientific applications has brought about the bid to develop security standards, among which are digital signature and encryption standard for XML. Syntax and processing specification of XML signature describes an XML syntax for representing the association between cryptographic signatures and XML documents or other electronic resources. Procedures for computing and verifying XML are included in the specification. Also, the XML signature defines mechanisms for countersigning and transformations known as canonicalization, which ensures that two instances of the same text produce the same digest for signing even if their representation differs slightly. it also defines XML vocabulary and processing rules for protecting the confidentiality of XML documents in whole or in parts and non-XML data as well.

16. Access Control for the Web and Mobile Applications

One unique challenge to database security is publicly accessible Web applications. The system includes those responsible for handling sensitive or private information and includes social networks, mobile application API servers, and e-commerce transaction platforms. Electronic commerce (e-commerce) environments are characterized by any transactions that are done electronically which requires elaborate access control policies that go beyond traditional DBMSs. The access control mechanism must be flexible enough to support a wide spectrum of heterogeneous protection objects. To prevent data breaches in these systems, a first requirement is a comprehensive information security policy that goes beyond the technical access control mechanisms found in mainstream DBMSs. The second requirement is the support for content-based access control. This allows one to express access control policies that take the pro protection object content into account. Accessing control policies must allow the inclusion of conditions based on the object content. The third requirement is related to the heterogeneity of subjects, which requires access control policies based on user characteristics and qualifications rather than on specific and individual characteristics [3].

17. SQL Injection

With a SQL injection attack, your database security is compromised and allows for bad actors to gain access to your database and the sensitive information in which it contains. The SQL injection attacks are conducted in usually in one of three ways by using malicious SQL statements in an entry field: SQL manipulation A manipulation attack, changes an SQL command in the application. Code Injection: This type of attack attempts to add additional SQL statements or commands to the existing SQL statement by exploiting a computer bug, which is caused by processing invalid data. Function Call Injection: in this type of attack, a database function or operating system function call is inserted into a vulnerable SQL statement to manipulate the data or make a privilege system call [4].

17.1. Risks Associated with SQL Injection

Risks that can be associated with SQL injections can be detrimental to a database as database fingerprinting, which, an attacker figures out what type of database is being used to tailor the SQL injection attack for a higher success rate.

Denial of Services attack is also possible as well as bypassing authentication which is a necessity for database security as you want to verify all users who have access to a database. Remote commands are also a concern as someone can launch a SQL attack from anywhere in the world which can make tracking the source of the attack.

Hackers may exploit database vulnerabilities such as misconfigured security settings, SQL injection, buffer overflows to obtain unauthorized access to database or inject malicious code. Therefore, it is necessary to keep database software and system update with the latest security patches for mitigating risks from known database vulnerabilities and exploits. During these times, DBAs may have to face some challenges in patch management, such as accident delays, compatibility issues, or downtimes.

17.2. Protection Techniques against SQL Injection

There are ways to protect against SQL injection such as Bind Variables which is using parameterized statements. The use of bind variables (also known as a parameter) protects against injection attacks and also improves performance as well as Filtering input (Input Validation). This technique can be used to remove escape characters from input strings by using the SQL Replace Function [5].

For detecting suspicious behaviors, unauthorized access attempts, and compliance violations, DBAs should pay more attention to monitor database activity and perform periodic audits. As a result, specialized tools and expertise may be required.

18. Statistical Database Security

With a statistical database, security is of a higher level of importance. Within a statistical database, it is not permissible to retrieve data that can be used to identify individuals such as an entity by name, income, or race for example. To protect individual data, queries only target statistics such as SUM, AVERAGE, and MAX. Queries that can pull individual data are not permitted; this can include: NAME, SALARY, AND, AGE. Database managers have to implement controls to prevent inferences and to ensure the confidentiality and the prevention of retrieval of personal information. What is inference? Inference, in relation, is statistical databases, is taking a query to obtain personal information that should otherwise not be obtainable.

The book (Pearson Fundamentals of Database Systems) gives the following example of inference:

“Q1: SELECT COUNT

(*) FROM PERSON

WHERE <condition>;

Q2: SELECT AVG (Income) FROM PERSON

WHERE <condition>;”

“Now suppose that we are interested in finding the Salary of Jane Smith, and we know that she has a Ph.D. degree and that she lives in the city of Bellaire, Texas. We issue the statistical query Q1 with the following condition:

(Last degree = “Ph.D.” AND Sex = “F” AND City = “Bellaire” AND State = “Texas”)

If we get a result of 1 for this query, we can issue Q2 with the same condition and find the Salary of Jane Smith. Even if the result of Q1 on the preceding condition is not 1 but is a small number—say 2 or 3—we can issue statistical queries using the functions MAX, MIN, and AVERAGE to identify the possible range of values for the Salary of Jane Smith.”

There are multiple ways that database managers can prevent inferences from happening and to ensure that specific information cannot be retrieved, or an inference cannot be deduced. A database manager, for example, can implement a threshold on queries; this means that you cannot execute a query if your query falls below certain restraints, such querying only a population’s specific age or a range of ages such as “31” or “25 - 30” range or querying an AGE, CITY, and SEX. We can also limit the number of queries that can be conducted on the same population. For example, a Q1 that pulls AGE and Degree type of a city population, then executing a Q2 on those results to narrow down to information that is trying to be obtained. Putting limits on the number of queries that can be executed is a good prevention method of obtaining personal information on specific entities as well as inferences as you cannot keep queering to get down to a specific result that you are intending to obtain [6].

Noise or slight inaccuracies can also be implemented into the statistical database as to obscure specific results, this could include having certain queries have salaries with a slightly off dollar amount or having the queries return "12 results" when in reality, there might only be 10.

19. Other Issues in Statistical Database Security

Other issues can include the use of trackers: "Scholer's individual tracker It permits calculating statistics for arbitrary query sets, without requiring preknowledge of anything in the database." (Dorothy E. Denning, Peter J. Denning, 1979. The tracker: A threat to statistical database security. ACM Trans. Database Syst. 4, 1 (March 1979), 76-96. DOI: https://doi.org/10.1145/320064.320069) [7]

Output perturbations and data perturbations can be used for the security of statistical database, and these methods involve introducing "noise" or slight inaccuracies into a database. These methods can also attribute to a data-mining bias [6].

20. Flow Control

In database systems, it is essential to examine how information flows through a transaction or program. Flow control monitors the flow or distribution of information among accessible objects. An example of flow occurs when information flows from when object X is read and then written into object Y. Flow controls check that information contained in certain objects does not flow explicitly or implicitly into less protected objects. The concept of flow control first appeared in the 1970s. In order for flow control to work some form of security class must be incorporated. Information can only be exchanged if both users exist in security classes that have the same privileges between the sender and receiver of the information. When a website leaks confidential customer information like a hashed or un-hashed password of one user to another user via a result of a web form is one example of the type of flow that must be avoided to maintain the integrity and confidentiality of a database [8].

In order to maintain flow control of a database, a flow policy must be developed. The flow policy will define the channels which information is allowed to travel through. One of the simplest examples of flow policy is made up of two security classes of Confidential (C) and Nonconfidential (N). In this policy, all information flows are allowed except for when information flows from C to N. The policy works by using access control mechanisms to check user's authorization for resource access. The policy dictates only granted operations are allowed to be executed. In order to assign a security class or clearance to run each program extended access control mechanisms are required to enforce access. A program is only allowed to read a memory segment if its clearance is as high as that particular memory segment. A program is only allowed to write a segment if its security class is as low as the memory segment. This helps prevent the flow of information from a high to low class. A real-world example of this is a military program with a secret clearance that is able to read objects that are either unclassified or confidential, but the same program is only allowed to write objects that are secret to top secret.

Flows can be classified into one of two types. First, a flow that occurs as an intended consequence of an assignment instruction is known as an explicit flow. An example looks like $Y: = f(X_1, X_n)$. The other type of flow occurs as a consequence of assignment instruction. An example looks like f $(X_{m+1}, \ldots, X_n)$ then $Y: = f(X_1, X_m)$. The goal of flow control is to ensure that only authorized explicit and implicit flows to occur. This is accomplished by constructing a set of rules that when met will allow information to flow securely through the application. These rules establish flow relations amount classes and information about how information can flow in a given system.

21. Covert Channels

Covert channels are examples of channels that allow any transfer of information that violates the policy or the security of a database or application. In many cases, improper means may be exploited to allow higher classified information pass to a lower classification level. There are two types of covert channels that exist in systems. First, there are timing channels that allow for information to be conveyed due to the timing of the event or process. Second, there are storage channels that allow information to be conveyed through accessing system information that the user typically should not have access to. Unlike Timing channels, temporal synchronization is not required to exploit storage channels. Both distributed databases and operating systems allow enough control over multiprogramming of operations that allows the sharing of resources without the possibility of encroachment of one program or process into another's memory or other system resource making timing based covert channels very unlikely. Covert channels are not a major problem in well-implemented robust database systems, but certain schemes can be created by clever users that could implicitly transfer information in a way that violates the confidentiality of the information. One best practice to avoid covert channels is to disallow programmers access to confidential information after the program has become operational [9].

22. Encryption and Public Key Infrastructures

Although flow control is a strong measure to maintain the security of a database system, it will not be able to prevent every attack. Since flow control is not guaranteed it is recommended that the information is disguised in the event that it falls into the wrong hands. The best way to accomplish this task is by using encryption on the data in the database. The process of encryption converts data into ciphertext. The ciphertext is a state that the data cannot be easily understood by unauthorized parties. The textbook defines ciphertext as the encrypted data and the plaintext as data that can be understood or acted upon without the need to employ decryption on it. The process of decryption is described as the process of transforming ciphertext back into plaintext.

The process of encryption works by first applying the encryption algorithm to the data with the special corresponding encryption key. After this process is successfully completed the only way to recover the original data is by using the proper decryption key to unscramble the data. A few different encryption standards exist that use different methods of how the algorithm scrambles the data into ciphertext. The two most commonly used encryption standards are Data Encryption Standard (DES) and Advanced Encryption Standard (AES). Data Encryption Standard (DES) was developed by the US government but is widely accepted and used across the globe as a cryptographic standard. It provides end to end encryption on a channel between a sender and receiver. The two fundamental building blocks of the algorithm depend on substitution and permutation (transposition) of the data to create the ciphertext. The strength of the DES standard derives its strength from a direct result of repeating 16 cycles of both substitution and permutation on the original data. The plaintext or original data is encrypted into blocks of 64 bits. Despite this fact, the key can be any 56-bit number. The National Institute of Standards and Technology (NIST) developed the Advanced Encryption Standard (AES) because of concerns of the adequacy of the DES standard. The AES algorithm uses block sizes of 128 bits in comparison to the 56-bit block sizes used by DES. AES can use keys ranging in size of 128,192 or 256 in comparison to the key being only 56 bits. Since there are more possible keys for AES it takes a much longer time to crack compared to DES. Both Apple FileVault and Microsoft BitLocker use AES with a large key length as the standard for their encryption. AES with a large key length is also the default encryption standard on most modern operating systems. If a legacy system cannot run any modern encryption standards, TripleDES can be used for encryption. It is a variant of DES that reruns the DES algorithm on the data three times to strengthen the short key that des typically uses

Both AES and DES are considered Symmetric Key Algorithms. A symmetric key is a key that is used for both encryption and decryption. Information encrypted with a secret key can only be decrypted with the same key that it was encrypted with. Symmetric key encryption algorithms are also known as secret key algorithms. Since these secret key algorithms are mostly used to encrypt the content of messages they are also known as content-encryption algorithms. Using the same key allows for fast encryption and decryption but this convenience makes the encryption less secure since the secret key must be shared between two parties. One way is to share a key from a user-supplied password string. Sender and receiver will apply the same function to the identical string of user-supplied password. This process is referred to as a password-based encryption algorithm. The strength of the symmetric key depends on the size of the key used for encryption. Longer keys are tougher to crack than shorter keys.

Another popular method for encryption is called Public or Asymmetric Key Encryption. Public key encryption does not depend on operations on bit patterns instead it uses mathematical functions to employ its encryption. One advantage public key cryptography has over symmetric key cryptography is that the sender and receiver do not need to share encryption keys in a secure manner. Public or Asymmetric Key Encryption requires the use of a public and a private key for encryption and decryption of data. The public key is allowed to be transmitted in a non-secure way while the private key is never transmitted and must be kept a secret. Both keys are mathematically related, but it is extremely difficult to derive one key from another. A public key encryption scheme works by first feeding plain text into the encryption algorithm as input. The encryption algorithm performs various transformations on plain text depending on the combination of the public and private key that is used for encryption if the public key is used to encrypt text than only the private key can be used to decrypt it. The result of the encryption algorithm outputs ciphertext. The ciphertext will need to be included with the other key pair that was not used for encryption in the decryption algorithm to recover the original text. In order to send private messages between users, each user must create a private and public key pair. After both keys are generated, the user must place the public key in a public register or accessible file to make it public. When someone wants to send a message to a particular user, they will use that particular user's public key to encrypt the message. The particular user will then use their private key to decrypt the message from the other user. The most widely accepted and implemented approach to public key encryption is the RSA Public Key Encryption Algorithm. RSA combines the difficulty of determining prime factors, modular arithmetic and results from number theory to scramble data [10] [11].

23. Digital Signatures

Digital Signatures use encryption to provide authentication services in e-commerce transactions. The goal of a digital signature is to link a unique user with a particular body of the text. A digital signature is a string of symbols that is different each time it is used to prevent easy counterfeiting. Digital signatures are created from a function of the message that combines a user's secret number with a timestamp value to create a unique signature. In order to verify a digital signature, the user's secret number does not need to be known. The best way to implement digital signatures is through public key encryption techniques.

24. Digital Certificates

A Digital Certificate is a digitally signed statement that combines a public key value with the identity of service or person that holds the corresponding private key. Digital certificates are issued by a certification authority (CA). The person or service receives the certificate from a certification authority (CA) that they are subject to validate their identity. Digital certificates make it possible for third-party authentication, so every user does not have to verify each individual's identity every time they do business with someone. Information that is included in a digital certificate are certificate owner information such as a distinguished name (DN) and other information about the owner, public key of the owner, issue date of certificate, valid to and from dates of certificate, issuer identification and other information about certification authority (CA), digital signature of certification authority (CA). The digital signature of a certification authority (CA) is a message digest function that creates a digital signature to certify that the association between public key and certificate owner is valid.

25. Privacy Issues and Preservation

Data preservation is an important technique that needs to be utilized in database security. You need to simultaneously provide data privacy and data preservation. Providing proper data preservation methods would consist of maintaining data quality and avoiding central warehousing for vital information. Having a central location for all of your data would present security issues for example: If you are storing vital medical information in a single repository and it gets compromised, a malicious individual would have access to everything. Implementing backup and recovery strategies and ensuring data redundancy and fault tolerance are measures to minimize the risk of data loss when accidental deletion, corruption, or software and hardware failures exist.

However, there are ways to combat malicious data mining techniques and that is done by modifying, perturbing, anonymizing, injecting noise, removing identity information associated with the data. There is still a major concern in data quality because implementing too many modifications will affect the overall quality of data. Database privacy is one of the biggest areas of focused because this ensures trust. Privacy involves implementing access to control-based techniques. User identities, profiles, permissions, locations to the data, and credentials all need to be kept secured. This isn't just a necessity at the individual level either, maintaining database privacy and preservation is also a large problem for hidden database owners as well. The fact that many web databases use restrictive form-like interfaces that allow users to execute search queries, these database owners would want to maintain security for aggregate information to protect their business secrets and homeland security.

26. Challenges to Maintaining Database Security

Database security is a growing need and what's making it difficult to maintain is the added features. Traditionally, database security involved confidentiality, integrity, availability, and privacy, but now for your database to be truly secure and modern, it needs to maintain a high level of quality, completeness, timeliness, and provenance. Maintaining database security is a difficult task when you factor in you have to also maintain data quality. The challenge is that there isn't much research out there on maintaining data quality, integrity-checking semantics, or even tools that help with the assessment of data quality. Future research is even needed for recovery tools and making sure it repairs lost/scrambled data accurately. Since there is a growing need, there need to be techniques in place, and organizational solutions, that can verify the quality of data.

27. Intellectual Property Rights (IPR)

This is another area where research is heavily needed. Data is generated by just about everyone, every day, not just from individuals by organizations as well. Since we do have data everywhere, organizations are also concerned with the legal troubles that this could bring. To help combat their concerns, watermarking techniques for relational data were introduced. "Digital watermarking is to actively protect content from unauthorized duplication and distribution by enabling provable ownership of the content."

28. Database Survivability

The last challenge that will be touched on in this topic is the survivability of a database. This is another important area where there hasn't been much research. Database survivability is the ability of a database to continue its functions after a catastrophic failure that causes reduced abilities, or an attack. Even though database security helps with the prevention of attacks, database survivability needs to focus more on the active occurrence of an attack. The first step is confinement, confinement means taking immediate action to either neutralize or isolate the threat. Step 2 is damage assessment, and that is simply assessing the damage that was done. Step 3 is reconfiguration, this is the start of the recovery process while allowing for regular operation. Step 4 repairs, the repair is fully recovering lost and corrupt data and the start of normal operation. The last step is fault treatment, and this is identifying the weaknesses, and learning from the exploits to prevent future attacks.

29. Oracle Label-Based Security

The oracle labeled-based security provides security at the row level with access control features. It is an advanced way is of controlling access to sensitive data. It works by giving both the data and user labels. The user label signifies what information the user can access while the label containing the data indicates what level of sensitivity you to need to access it, as well as the ownership. Compartments are used to assist with the classification in the sensitivity of the data, groups are used to identify the organizations as owners in regard to the labels. "If a user has a maximum level of SENSITIVE, then the user potentially has access to all data having levels SENSITIVE, CONFIDENTIAL, and UNCLASSIFIED. This user has no access to HIGHLY_SENSITIVE data." Oracle labeled-based security is built on Virtual private databases (VPDs). VBDs are features of the Oracle Enterprise Edition that add predicates to user statements to limit their access in a transparent manner to the user and the application. VPD adds access control based on policies that enforce object-level access control or row-level security. VPD also provides an application programming interface (API) that allows security policies to be attached to database tables or views. Using PL/SQL, a host programming language used in Oracle applications, developers and security administrators can implement security policies with the help of stored procedures.

30. Summary

There are many different areas of database security all of which are in important. This paper helps present future research areas as defined in the book of "*Fundamentals of Database Systems*" as well as other research papers. Different threats to databases such as SQL injection attacks, or any attack that can compromise a database will cause loss of integrity, availability, confidentiality, privacy, and ultimately trust. We have introduced different control measures and then discussed mechanisms for granting and revoking privileges in the relational database system. We have also specified security mechanisms including discretionary access control, mandatory access control, role-based access control, and XML access control. Furthermore, we offer specific preventive measures to threats, such as SQL injection in the database. Even for statistical databases, there are security problems, so it is important to pay attention to flow control and covert channels. To keep data encrypted, we summarize encryption and symmetric key and asymmetric key infrastructure schemes and discuss digital certificates as well. We have also discussed Privacy-preserving techniques like avoiding a large central repository of data and implementing access control to ensure privacy. To maintain database security, we research current challenges that involve data quality, IPR, and database survivability. We have also provided some detail in Oracle label-based security.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Wang, H. (2012) Security and Privacy for Database Systems. *Proceedings of the Twenty-Third Australasian Database Conference* (*ADC* 2012), Melbourne, 31 January-3 February 2012, 5-6. https://dl.acm.org/doi/10.5555/2483739.2483741

[2] Thuraisingham, B. (2007) Security and Privacy for Multimedia Database Management Systems. *Multimedia Tools and Applications*, **33**, 13-29. https://doi.org/10.1007/s11042-006-0096-1

[3] Ferrari, E. and Thuraisingham, B. (2024) Security and Privacy for Web Databases and Services.

[4] Majumder, J. and Saha, G. (2013) Analysis of SQL Injection Attack. *International Journal of Computer Science & Informatics,* **2**, 2231-5292.

[5] William, G.J. Halfond, J.V. and Alessandro, O. (2006) A Classification of SQL Injection Attacks and Countermeasures. https://faculty.cc.gatech.edu/~orso/papers/halfond.viegas.orso.ISSSE06.pdf

[6] Adam, N.R. and Wortmann, J.C. (1989) Security-Control Methods for Statistical Databases: A Comparative Study. *ACM Computing Surveys*, **21**, 515-556. https://doi.org/10.1145/76894.76895

[7] Denning, D.E. and Denning, P.J. (1979) The Tracker: A Threat to Statistical Database Security. *ACM Transactions on Database Systems*, **4**, 76-96. https://doi.org/10.1145/320064.320069

[8] Almutairi, A.H. and Alruwaili, A.H. (2012) Security in Database Systems. *Double*

Blind Peer Reviewed International Research Journal, **12**, 9-13.

[9] Elmasri, R. and Navathe, S. (2011) Database Security-Concepts, Approaches, and Challenges. *IEEE Transactions on Dependable and Secure Computing*, **2**, 1-19.

[10] Rivest, R.L., Shamir, A. and Adleman, L. (1978) A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, **21**, 120-126. https://doi.org/10.1145/359340.359342

[11] He, J. and Wang, M. (2001) Cryptography and Relational Database Management Systems. *Proceedings* 2001 *International Database Engineering and Applications Symposium*, Washington DC, 16-18 July 2001, 273-284. https://dl.acm.org/doi/10.5555/646290.687060