Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.1247 – 1253

RESEARCH ARTICLE

A DYNAMIC APPROACH FOR DATABASE SECURITY

Mr. Mahesh Singh[1], Ms. Alka[2]
[1]Asst. Professor, Department of Computer Science Engineering, AITM, PALWAL, INDIA
[2]Student, AITM, PALWAL, INDIA
[1] mahesh100nucs@gmail.com, [2] alka2513@gmail.com

Abstract

The growth of the Database domain and the wide Internet has been revolutionized over many years, a large number of people interact through information over the internet. This rapid change and the cost effectiveness of new evolving technologies are providing large opportunities for developing distributed database systems. These large-scale systems are made up of various interacting components, each of which is well encapsulated. However, this rapid growth has also brought many security issues, as data is now made available on the open Internet is delicate.

Keywords: DataBase, DOS (Denial of Service), Access Control, dynamic Certificate, Attacks

1. INTRODUCTION

1.1 DATABASE SECURITY

As Database domain increasing growth and the wide Internet has been revolutionized over many years, a large number of people interact through information over the internet. There is strong need for information security because of many factors.

Database security concerns the use of a broad range of information security controls to protect databases (potentially including the data, the database applications or stored functions, the database systems, the database servers and the associated network links) against compromises of their confidentiality, integrity and availability. It involves various types or categories of controls, such as technical, procedural/administrative and physical. .

Security risks to database systems include, for example:

- Unauthorized or unintended activity or misuse by authorized database users, database administrators, or network/systems managers, or by unauthorized users or hackers (e.g. inappropriate access to sensitive data, metadata or functions within databases, or inappropriate changes to the database programs, structures or security configurations.
- Malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services;
- Overloads, performance constraints and capacity issues resulting in the inability of authorized users to use databases as intended[1]

The present invention is for developing a secure process for the managing the database within the organization. When database is deployed to store financial and personal data, the real need for high security is felt. This invention uses the concept of Dynamic certificates to provide security to database. This DB-Certificate process will provide a mechanism to assign digital certificates to its user according to their designation/ level in the organization. These certificates will contain the types and list of queries which the user can execute on the database. This will help in implementing security principle as well as granting and revoking privileges to/from the accounts. Only after verifying the authenticity of the certificate, the authorized part of the database can be accessed by using the list of the queries attached with the certificate.

The security protocol is comprises of a client machine to request and use certificates while communicating across the network and a server machine that manage issuance and the maintenance of security certificates. The server machine receives request for a certificate from a client machine. There are a set of some predefined protocols to process these requests. A central certification authority (CA) is a trusted party which provides certificates when receive requests from the client. The CA has a set of predefined policies which are implemented to generate these certificates. These set of policy engines has at least one policy which is configured as a software element, for e.g. a Java bean can be used to perform the various distinct functions defined in the policy and generates notifications as a response. The certificate is generated dynamically each time when the client requests. This prevents the certificates from hackers. Each level of user has a certificate which will contain the types and list of queries which the user can execute on the database. This will help in implementing security principle as well as granting and revoking privileges to/from the accounts. Only after verifying the authenticity of the certificate, the authorized part of the database can be accessed by using the list of the queries attached with the certificate.

The dynamically generated digital certificate validation method of a client connectable in communication with a host is provided. The very first connection is made up with the host to establish data communication within the host. Also, a request for a certificate validation result is sent back to the host. Then a file containing the requested certificate validation result is imported from the host and the imported file is stored locally for later retrieval of at least the requested certificate validation result.

1.2 ARCHITECTURE FOR DATABASE SYSTEMS

Software systems generally have an architecture, i.e. Possessing of a structure (form) and organization (function). The former describes identifiable components and how they relate to one another structurally; the latter describes how the functions of the various structural components interact to provide the overall functionality of the system as a whole. Since a database system is basically a software system (albeit complex), it too possesses an architecture. A typical architecture must define a particular configuration of and interaction between data, software modules, meta-data, interfaces and languages [2].

The architecture of a database system determines its capability, reliability, effectiveness and efficiency in meeting user requirements. But besides the visible functions seen through some data manipulation language, a good database architecture should provide:

- Independence of data and programs
- Ease of system design
- Ease of programming
- Powerful query facilities
- Protection of data

As new computing methods have evolved, different methods of transferring the data between the database systems and the end users have been also evolved. For database-backed up systems, there are three most common architectures as follows:

- A direct link to the computer which performs all the work.
- A client/server (two-tier) architecture
- A thin client (three-tier) architecture

In the first architecture, the database(s), database software, program code, and all other resources located on one local machine. as shown below.

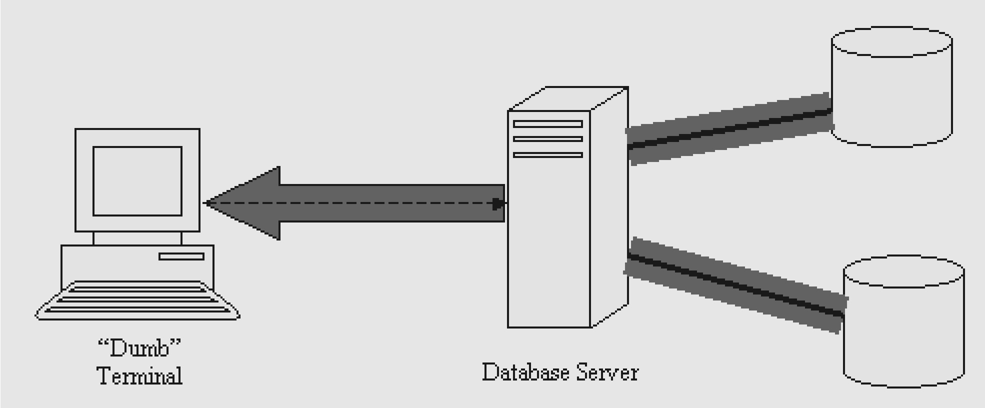

Figure 1: Direct connection to database server

In the second architecture, the database server and the database and other resources are placed at one location while the code for application is placed on a client machine or on their personal computer. In this architecture, queries are carried to the server machine and the resultant data is carried back to the client's machine by pre-defined queries. Queries are processed on the database server, while rest part is done on the workstation. The processing at workstation mostly includes managing the user interface and display, and the application processing which is independent of the database. This architecture is shown below.

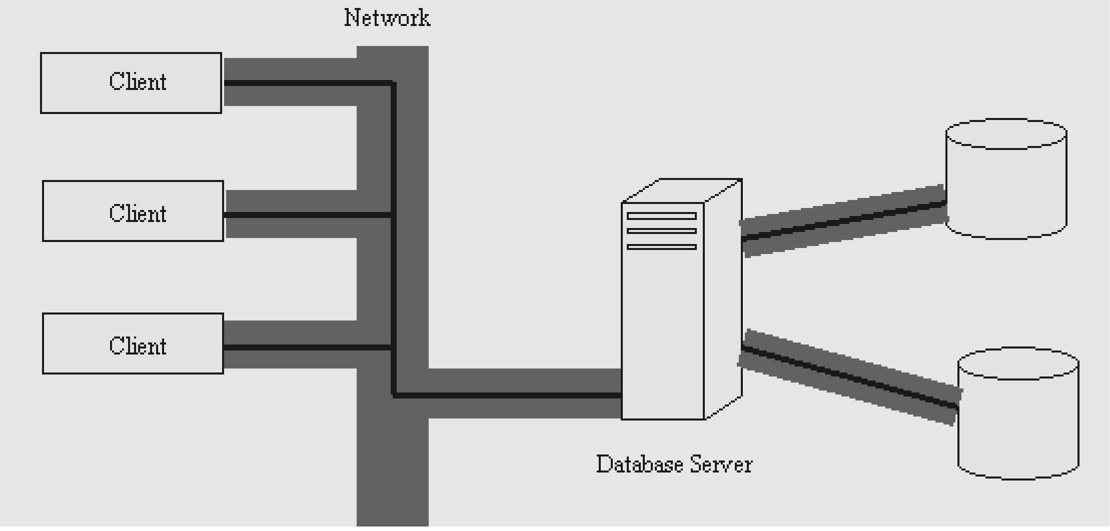

Figure 2: Client/Server architecture

The latest type of architecture is more efficient and uses a low capacity workstation as a first tier. The workstation does not run the application really. It only manages display of the GUI and taking user inputs. It processes the requests by downloading whole code onto the client machine. The second tier in this is the Application Server. It executes and holds the applications using some programming language and communicates with the workstations. The applications that executes on this server machine communicate with the third tier and server database by using the protocols for the databases. Where security concerns, the three-tier architecture is very useful in many ways. It requires higher security requirements on the application server, minimal security requirements at the client side and changing number of security requirements at the back end server databases.

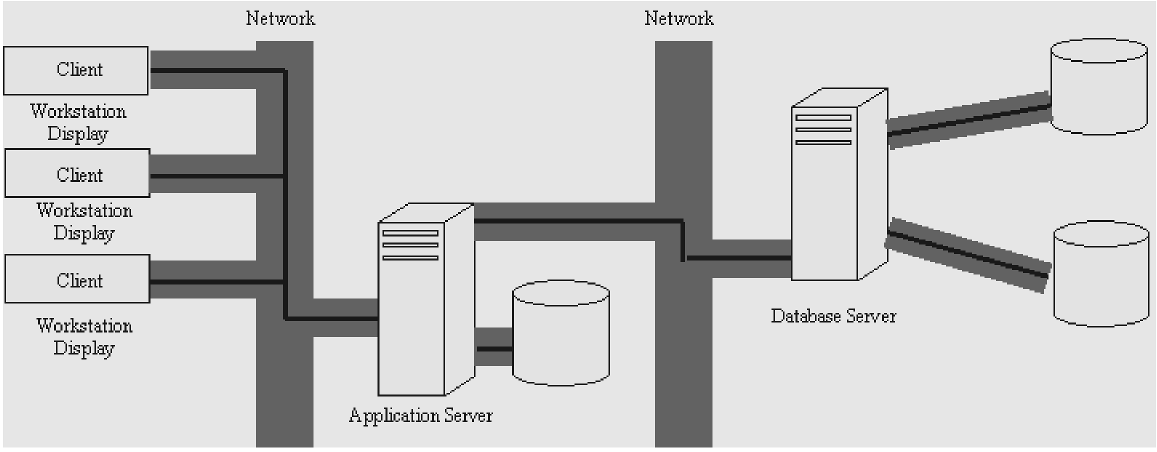

Figure 3: Slim Client architecture

In general, the goals of database security are:

- Confidentiality and secrecy: Data should not ever be revealed to anyone who is not authorized to access it.
- Authentication, accuracy and integrity: It means that data cannot be modified maliciously or corrupted intentionally. Authenticity provides an easy way to user verify the original location of the data.
- Recoverable and availability: Systems should continue working, and the lost data could be recovered easily, efficiently and in the original form.

2. ATTACKS ON DATABASE

Two kinds of attack can be made to the databases; Physical attack and the logical attack. Physical attacks can include forced disclosure of sensitive information like passwords, demolition of storage devices in system, complete power failure, and theft of secured information. To prevent these kinds of attacks, common way is to limit the access to all the storage devices. Keep the backup and recovery procedures safe.

While logical threats are intentionally or unauthorized access to sensitive information. This is usually done through software. Logical threats can leads to denial of service (DOS) attack, disclosure of sensitive information, and tempering of data.

2.1 Insider Threat

Corrupt authorized user can harm the system. These kinds of users can legitimately access the confidential information and misuse it. This confidential information can be exposed electronically, through print outs to the outer organizations. To prevent information from within the organization is very difficult. Mandatory access controls are required to hide data from unauthorized users. This kind of attack is usually taken care by limiting the number of users at different levels of access. Complicated procedures can also be used to address them

2.2 Login Attacks

Another more popular way to damage a database is to login successfully to the system as a valid user. Passwords can be stolen physically or by monitoring network traffic continuously for login data. Also, password lists can be accessed which are stored in the files of operating system. If password is easy to guess, there are full chances of system damage. Restrictions on passwords can be done to prevent from loss, but it does not completely solve the problem. The database administrator must employ encryption mechanisms and authentication procedures to prevent this kind of attacks.

3. ACCESS CONTROL MODEL

Various Access control models address above stated goals are as follows:

3.1 Database Access Control Models

Access control models were developed to mainly address the affairs of data secrecy, availability and confidentiality. The models can be classified as either recent or traditional. Further traditional access control models, are broadly classified as Discretionary Access Control (DAC) models and Mandatory Access Control (MAC) models. Now newer models comprise of processes such as Task-based access control (TBAC) models and Role-based access control (RBAC) models. These models address the security requirements for large range of applications. Main differences among these models are as below:

3.2 Discretionary Access Control (DAC) Model

This model is based on the concept of privileges (or access rights) for the data objects and the mechanisms to grant or revoke permissions and give users such privileges. These access rights allow a user to access some data objects in a particular manner (e.g., writing or reading the data). All these subjects and objects in the system are computed and their access authorization rules for all subjects and objects are specified. The subjects can be users, group of users, or group of processes which can act other subjects. If any subject owns an object, the subject has all permission to grant or revoke access rights for objects and to other subjects at his or her caution. DAC policies are adaptable and the most broadly used for Web-based applications. But these policies do not provide good security assurance. For e.g., DAC allows to copy data from one object to another object, and this can result in providing access to a copy of data to all those users who doesn't have permission to access the original data.

3.3 Mandatory Access Control Model (MAC)

Mandatory access control (also called security scheme) is based on system-wide policies that cannot be changed by individual users. It is used to enforce multi-level security by classifying the data and users into various security classes or levels and then implementing the appropriate security policy of the organization. Thus, in this scheme each data object is labeled with a certain classification level and each user is given a certain clearance level. A given data object can then be accessed only by users with the appropriate clearance of a particular classification level[3].Mandatory access control model is based on system-wide policies which cannot be changed by individual end users. In this model, security classes are assigned to every data object, and then each user is assigned permission for a security class. Also, rules are imposed on writing and reading the database objects by end users. Most important goal of this access model is to control the flow of information in order to ensure integrity and confidentiality of the data, which cannot be addressed by above DAC models. For e.g., In Defense applications to ensure information confidentiality, a MAC model can be executed using a multilevel security mechanisms which uses no write-down and no read-up rules, also known as Bell-LaPadula restriction.

3.4 Role-based Access Control (RBAC) Model

Role-based access control (RBAC) model is gaining increasing attention as a more generalized perspective to access control because they provide many well-recognized advantages over other traditional ACLs. In fact, they have been regularized by NIST. RBAC allows access based on the job title. For example, a human resources specialist should not have permissions to create network accounts; this should be a role reserved for network administrators [4]. Roles are captured from a user's functions and responsibilities within a corporation. A role-based model supports directly arbitrary and organization-specific security policies. However, ACLs are tied only to some particular objects and the difficult task is their maintenance for system administrators in an organization.

4. PROPOSED DESIGN AND IMPLEMENTATION

"Securing database using dynamic certificates" is an approach that provides a mechanism to assign digital certificates to its user according to their designation/ level in the organization. These certificates will contain the types and list of queries which the user can execute on the database. This will help in implementing security principle as well as granting and revoking privileges to/from the accounts. Only after verifying the authenticity of the certificate, the authorized part of the database can be accessed by using the list of the queries attached with the certificate.

The hierarchy is made for users according to their designation or power like, administrator, employee and outside users.

According to the role of a user, his/her access rights to fire a query and to operate the database system are created. For e.g., An administrator has full access rights to fire any query like delete, update, create, joins etc. He also has rights to run complex queries using WITH, temporary tables and to create and run stored procedures, Table value functions, Scalar value functions, Inline functions. He also has authority to create schema diagrams, to inspect the elements and schemas created by other users.

On the other hand, employees of the organization may have limited access rights to create database or tables and schema diagrams. He is also allowed to take the backup of the database, but he does not have permission to run delete and update query.

Last but not the least; external users have very limited access to the database system. They can perform read operations but not the write operations. Also their security breach will be alarmed to the administrator.

These access rights are designed and formulated by the help of dynamic certificates. These certificates checks whether the user is allowed to access the database system by matching the fired query with its digital certificates.

4.1 Database Design

We have developed a hypothetical database application that is used as a real-world model for database security implementation. Our imaginary company is named the Alby Company. We have used our own designed database system name given Simu Database as our database server having a Web-based user access. This newly created database server needs to be made secure using different access rights for each group of users.

These access rights are bind with the digital certificates. These digital certificates are generated dynamically at the point for query fire. When a particular user fire a query, his dynamic certificate which has his assigned access rights are bound is generated. The user is allowed to access that particular query only if his certificate is verified, otherwise, the access is denied.

4.2 Algorithm Design

We have used a very powerful and secure language Java to design the algorithm for generating dynamic certificates and for checking the access rights against these generated certificates. Only after validating his/her identity and checking of their certificates, users are allowed to access the database system.

To perform these operations, an algorithm is designed in Java, Hibernate and Struts.

Hibernate. It is an open source relational object mapping tool for Java. Hibernate allows user to develop more persistent classes.

Hibernate is an object oriented query language which is responsible for mapping Java classes to the database tables and also from Java data types to the SQL data types. It also provides data retrieval and query processing. It supports various types of queries like: polymorphic queries, native SQL queries, composite queries, etc. It provides the high performance to the database.

4.3 Algorithm Steps:

1. An algorithm is generated using Java, Hibernate. This algorithm generates random key forthe dynamic certificates.
2. This dynamic certificate is generated each time when user tries to access the database system.
3. A hierarchy of users is created as administrator, employees, outside users.
4. Each level of users is assigned some access rights to access the database.
5. The queries which he/she is allowed to execute are bind with the dynamically generated certificate.
6. Now, when the user fire some query and tries to access the database system, a digital certificate is generated.
7. This certificate generates a random key at client's side.
8. If this key got matched with the key at server side, only then user can proceed further. Otherwise, user's access to database system is denied.
9. Next, if the keys match, then, the queries are checked which are bind with the certificate.
10. If the user is firing an authenticated query, only then he/she can access the database. Otherwise, an alert will be generated to deny the access and to inform administrator about this error.

This process is depicted in the below figure:

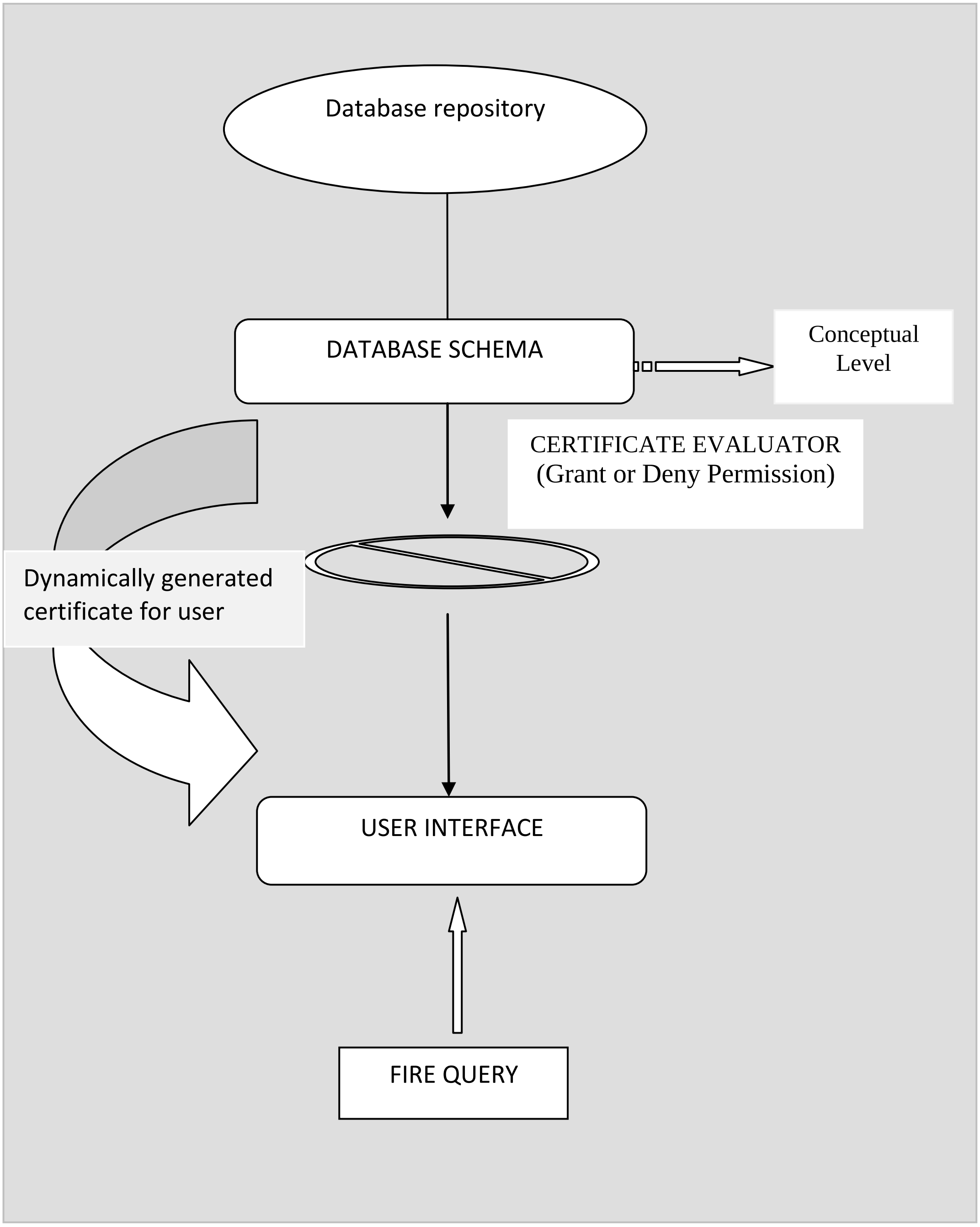
Database repository
DATABASE SCHEMA
Conceptual Level
CERTIFICATE EVALUATOR (Grant or Deny Permission)
Dynamically generated certificate for user
USER INTERFACE
FIRE QUERY

Figure 4: Algorithm representation

5. DISCUSSION AND FUTURE WORK

There cannot be only one complete solution for securing database and there are lot many issues which are tedious to remove. Any organization wants a secure database system then, they must secure the overall environment. This includes securing communication channels throughout the network, user access control methods, encrypted queries to the database system and securing the database system itself, and all those applications which are used to access the database server. A good combination of secure hardware and software can make the modern database servers more secure.

There are many techniques are available to provide security to the web servers, web browsers, network which are discussed above. In addition to these techniques, the database security levels can vary from organization to organization. Therefore choosing an appropriate design and access mechanisms are critical for securing database systems in the organization. Security through digital and dynamic certificates can be applied to any organization whether private or government. Right decisions for formulating access rights can be made user wise within the organization.

In the future, there are some areas which can help in enhancing the security of database system are:

- Encryption of data and queries.
- User training for accessing database system and to choose strong passwords for their systems.
- Communication layer can be made more secure using enhanced security mechanisms or by combining various security

methods.

- Audits can be created to maintain the logs for every user.
- Daily access reports can be sent as a mailer.

6. CONCLUSION

Various security problems in databases are presented. The database is not only be harmed by the unauthorized user in the internet but can also be misused by the legitimate user within the organization. Various security methods are available to protect database system from external users in the network. But to protect the database system from insiders, we have proposed a solution i.e. through the use of dynamic certificates.

The hierarchy is made for users according to their designation or power like, administrator, employee and outside users. According to the role of a user, his/her access rights to fire a query and to operate the database system are created. For e.g., An administrator has full access rights to fire any query like delete, update, create joins etc. He also has rights to run complex queries using WITH, temporary tables and to create and run stored procedures, Table value functions, Scalar value functions, Inline functions. He also has authority to create schema diagrams, to inspect the elements and schemas created by other users.

REFERENCES

[1] http://en.wikipedia.org/wiki/Database_security
[2] http://coronet.iicm.tugraz.at/Dbase1/scripts/rdbh12.htm
[3] http://my.safaribooksonline.com/book/databases/9788177585674/database-security/ch14lev1sec4
[4] http://www.evolllution.com/media_resources/cybersecurity-access-control