PostGRE Database Server and Overview on its Various Encryption Methods

Anshul Bansal
UG Students
Department of Computer Engineering
MES College of Engineering, Pune-411001, Maharashtra, India

Durva Deshmukh
UG Students
Department of Computer Engineering
MES College of Engineering, Pune-411001, Maharashtra, India

Shreyans Kering
UG Students
Department of Computer Engineering
MES College of Engineering, Pune-411001, Maharashtra, India

V. P Vimal Dev
UG Students
Department of Computer Engineering
MES College of Engineering, Pune-411001, Maharashtra, India

Shraddha R. Khonde
Assistant Professor
Department of Computer Engineering
MES College of Engineering, Pune-411001, Maharashtra, India

Abstract

Large amount of data is generated through different applications on a daily basis. Databases are used to store this data as per the requirements. This data contains a lot of personal and confidential data that is needed to me protected from being misused. The best way to protect this data is by encrypting it. In encryption what we do is basically scramble the contents of a respective field of the database with the help of a key. Encryption can serve as a mean to achieve flexibility and privacy of data can also be achieved which is very essential business environments. This paper presents different encryption methods used to encrypt different fields in a PostGRE database.

Keywords- One Way Encryption, Two Way Encryption, Decryption, Hashing algorithm, pgcrypto methodology

I. INTRODUCTION

An object-relational database such as PostGRE provides additional “object “features where significance and prominence is given to flexibility, adaptability and meeting the rules of standards. The primary function of the database server is to securely store data in itself and when third party applications request for its retrieval give permission for the same. With large number of concurrent users operating on the database PostGRE can handle loads of work that can range from application pertaining to small single machine to large numbers of applications from Internet users. For Web hosting the default database is PostGRE SQL.

To protect data from exposure due to larceny of the database server, unethical and illegitimate admins, and unreliable networks PostGRE database provides encryption methodologies at different levels and facilitates various flexibility measures for the same. To shield crucial and important data subjected to records from medical fields or transactions related to financial purposes, encryption is used for the same.

II. ENCRYPTION METHODOLOGIES

A. Encryption of Data Partition

In the Linux, using a "loopback device", encryption can be stratified on the mounted file system. Due to this, encryption of the integrated file system can be done, and its decryption can be done by the machine’s operating system. GEOM Based Disk Encryption or gbde is an equivalent facility on FreeBSD. If the storage drives or the whole computer system is embezzled then with the help of this mechanism reading of the unencrypted data is being prevented. When the mounting of the file system is done, the o.s. facilitates an unencrypted view of the data hence this mechanism does not shield it against intrusion when mounting of the file system is done. Frequently the key is stored on the anchor system where the disk is mounted; this encryption key has to be provided to the machine’s o.s. which will then enable the user for file system mounting.

B. Encrypting Passwords across a Network

Before sending the Password it is double encrypted on the client side using the MD5 authentication methodology and then it is sent to the server .When the connection to the database is established, MD5 authentication first encrypts the password based on the username and then it encrypts the password based on the arbitrary salt received by the client that is sent by the server. To the server over the network connection this double encrypted value is sent to the server. The main advantage of Double-encryption is that it precludes the password from being exposed, but also when the user tries to make connection to the database server later in time it prevents the other users on the same connection from applying the same password that is encrypted.

C. Encrypting Data across a Network

All data that is transmitted through the network connection i.e. the passwords, the queries for database, and the data returned from the network all are encrypted by SSL connection. The pg_hba.conf configuration file allows administrators of system to determine which hosts can use connections that are non-encrypted and which host requires connection that are SSL encrypted. Via SSL connection the clients can determine when they require for server connection. To encrypt and secure the transmission Stunnelor SSH can be used.

D. Client Side Encryption

The unencrypted data will never across the database server if the client encrypts the data in the case where the system admin cannot be trusted and is not reliable. Before sending the datasets to the server side it is first encrypted on the client side, and the database results that are encrypted at the client side before sending it to the server side, its decryption has to be done on the client side before it is capable of being to use.

III. PROPOSED WORK

Various Levels of encryption are provided by the PostGRE SQL from which the user can choose from. The contrib module namely pgcrypto provides basic, built in and advanced features. The basic and general rule while encrypting data is that the more intricate we make it to hide it from users, the more evident it becomes to lock ourselves out of our own data. After encryption not only does it makes it more challenging to view data, but it also requires even more assets to query the data and decrypt the database. With these thumb rules based on how sensitive the data is it is very crucial and important to select the database encryption methodology.

Basically there are two types of encryption first is one way and second is two way encryption. In the first method it doesn't matter about decryption of the data into user readable form, it is just that we want to certify that the user knows what the hidden classified text is. For Passwords this is generally used. In second method that is two way encryption, we want the feature that is data encryption as well as allowance of legitimate and authentic users to decrypt it into a readable and meaningful format. In this category datasets pertaining to SSNs and credit card data fall into this.

A. One Way Encryption

In PostgreSQL the by default encryption method built into it is MD5 and basically this is used when users want single way encryption and require a simple and basic encryption level. In MySQL there is a function called as PASSWORD, MD5 function is identical to this. If the users want to add anything beyond encryption, the user will require installing the module of pgcrypto contrib.

Most PostgreSQL install packages packs the pgcrypto contrib module including Windows Install package and the users can be install it into the database by the execution of the script in share/contrib/pgcrypto.sql of the users PostgreSQL install.

An enhanced security level over the md5 authentication is provided by crypt function in the pgcrypto module for one way encryption. Since no salts are provided by the MD5 encryption one can tell who have the identical and same password and all users with ditto identical passwords will eventually have same MD5 string. With crypt function the MD5 string will be different. To demonstrate the same lets we create a table in the database with two different users who have coincidently happen to choose the same password as the other has chosen.

In Fig. 1 Password test is preferred by both the users on the database the MD5 version of both the users' password is same but they have different cryptic passwords, when any user does the login procedure we perform this preliminary test.
Fig. 2 Demonstrates selection of password encryption methodology
Fig 3. Shows the system flow and where encryption is used

```
CREATE TABLE testusers(username varchar(100) PRIMARY KEY, cryptpwd text, md5pwd text);
INSERT INTO testusers(username, cryptpwd, md5pwd)
    VALUES ('robby', crypt('test', gen_salt('md5')), md5('test')),
        ('artoo', crypt('test',gen_salt('md5')), md5('test'));

SELECT username, cryptpwd, md5pwd
    FROM testusers;

 username |              cryptpwd              |              md5pwd

----------+------------------------------------+----------------------------
 robby    | $1$IOchfG/z$bZW1pRFA3wuvn6pAuD.Du/ | 098f6bcd4621d373cade4e832627b4f6
 artoo    | $1$84oZTXI/$yZ6wV5jhJo6aQYrTciMQR/ | 098f6bcd4621d373cade4e832627b4f6
```

Fig. 1: Comparison of MD5 and crypt

```
-- successful login
 SELECT username
     FROM testusers
     WHERE username = 'robby' AND cryptpwd = crypt('test', cryptpwd);
-- successful login
 SELECT username
     FROM testusers
     WHERE username = 'artoo' AND cryptpwd = crypt('test', cryptpwd);

-- unsuccessful login
 SELECT username
     FROM testusers
     WHERE username = 'artoo' AND cryptpwd = crypt('artoo', cryptpwd);

-- using md5
SELECT username
     FROM testusers
     WHERE username = 'robby' and md5pwd = md5('test');
```

Fig. 2: Selection method of password encryption

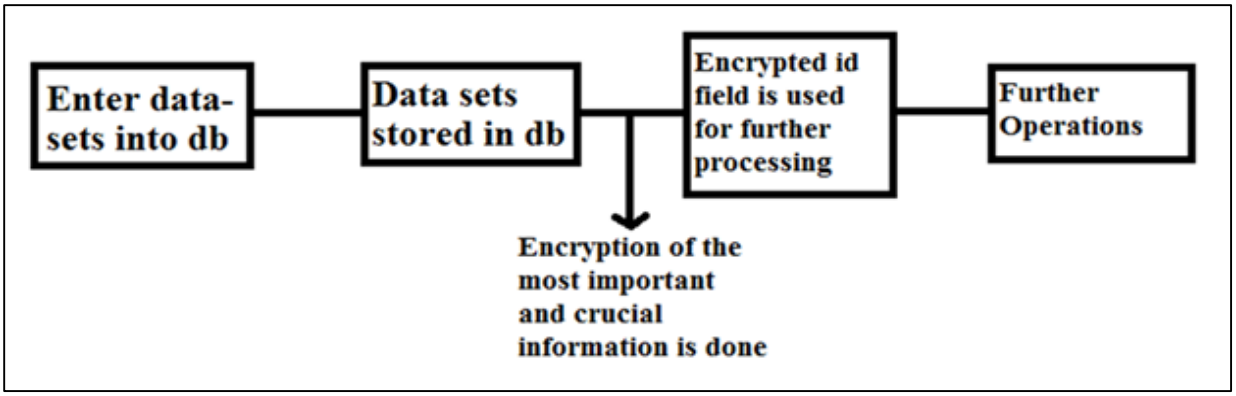

Fig. 3: System Flow and Encryption usage

B. 2 Way Encryption Decryption using PGP Encryption

For important and crucial data which is needed to be retrieved it doesn't matter what the information is, the only thing that matters is that only authorized users must be able to retrieve it . Credit card number, social security numbers and bank accounts numbers would account to such information.

The PGP encryption function in pgcrypto module is one of the most convenient and straightforward encryption modes. For Users PGP encryption modes comes in two types.

– Asymmetric (Public/Private) – This is called as the shazam mode of conduct because it can only be encrypted by public key and can only be read by private key.

– Symmetric – This is called as the shared secret web of trust. The encryption and reading keys are same.

A set of cryptography functions along with extensions such pgcrypto module is provided by PostGRE SQL.

IV. HASHING ALGORITHM WITH BIT SIZE

– MD5 = 128-bit hash value.
– SHA1 = 160-bit hash value.
– SHA224 = 224-bit hash value.
– SHA256 = 256-bit hash value.
– SHA384 = 384-bit hash value.
– SHA512 = 512-bit hash value

V. CONCLUSION

The paper reviews the implementation of various database encryption methodologies for the PostGRE database. The database encryption us a very crucial and important task. The paper outlines various approaches which can be applied for encryption such as one way and two way encryption that can be used for securing datasets.

ACKNOWLEDGMENT

The authors would like to thank MES College of Engineering, Pune for allowing us to use their resources.

REFERENCES

[1] http://www.dbrnd.com/2016/03/postgresql-best-way-for-password-encryption-using-pgcryptos-cryptographic-functions/
[2] PostgreSQL: Up and Running, 2nd Edition-A Practical Guide to the Advanced Open Source Database By Regina O. Obe, Leo S. Hsu