Five Security Features of RDBMS That Help Enterprises

The use of relational databases for storing certain unstructured documents have been recently challenged

Security Strength Of Relational Databases:

Over the years relational databases have been highly successful in protecting the enterprise data. Their inherent features like role based security, GRANTS and coupled with the fact that some one needs to have a high knowledge of the database design itself to really decipher meaning out of a relational database.

Unrelated to the above statement, we have seen a huge attack on large media enterprise last month resulted in the compromise of several of confidential documents which are now roaming freely in the public domain. While this article is nothing about who has done it and why they have done it, it is more about how old school thoughts on Relational database design could keep a company in a safe zone even after the network is hacked.

Over the last couple of years, the usage of Relational databases for storing certain unstructured documents have been challenged and some architectures moved towards file system based storage like, HDFS, Amazon S3, GFS, Azure BLOB etc.. While the above mentioned file systems are good for scalability, performance in a few situations, we cannot discount the value of relational databases in securely storing the data and ensure that an hacker will not gain the semantics of the data so easily even if the physical network and server security are compromised.

The following are some of the points in support of this argument.

RDBMS Design Is Complex:

Most of the File system based approaches deal with hierarchical storage and they don’t do normalization, which means while there is some efficiency in data storage they are also compromised much easier. Let us consider a HR Database about CEO Bonus for the last year.

In the file system storage.

/Year/Bonuses/Corporate/CEO.xxx.

Consider a typical and well thought out DB Design.

Employee Table (EmpId,......).

Code Table (Compensation Code (BONUS)).

Then the Transaction table which may link Emp Id, Compensation Code, and Value.

Also if IT departments take conscious decision not to use Friendly names for the tables and avoid foreign keys inside the database, i.e name the tables like T00001(F1, F2...) instead of EMPLOYEE_BONUS, and don’t keep the metadata documentation inside the database, then it would be almost impossible for any one from outside to hack this information, even when given a access to the database.

There is always a trade off between keeping a friendly names and meta data inside the database versus keeping them totally outside else where inside a development repository, most enterprises would be OK if a Developer struggles couple of days more in writing a query versus compromising all of their data to a hacker.

RBBMS Storage Is Proprietary:

One of the advantages of commercial databases like Oracle, DB2, SQL Server etc.. is that their internal architecture is highly proprietary making it difficult for other third party tools to decipher meaning out of them. For example the above databases support CLOB, BLOB storage types which are typically used for storing large amount of character and binary data, this means that these data can only be understood by an application written specifically for them and most times it is not easily shared with the outside world like the file system based documents that are shared freely in the case of recent hack.

Additionally most databases also support Encryption features, which makes deciphering the data outside of database context very difficult. For example the below are the notes about Transparent Database Encryption feature in SQL Server.

Transparent data encryption (TDE) performs real-time I/O encryption and decryption of the data and log files. The encryption uses a database encryption key (DEK), which is stored in the database boot record for availability during recovery. The DEK is a symmetric key secured by using a certificate stored in the master database of the server or an asymmetric key protected by an EKM module. TDE protects data “at rest”, meaning the data and log files. It provides the ability to comply with many laws, regulations, and guidelines established in various industries. This enables software developers to encrypt data by using AES and 3DES encryption algorithms without changing existing applications.

RDBMS Security is De-Coupled From OS, Network:

Most times it is the network and server security are compromised first before the database is conquered. In a good RDBMS design even the DBAs can be prevented from accessing the data, if they are carefully abstracted with the role based security, fine grained access control and other features like views. This means that the data can fully secured even if the network and server are compromised. Having said that most places there is a OS LEVEL authentication which gives complete control of database, but if thought out well, this link can be de-coupled and make your enterprise data more secure.

RDBMS Are Audited:

Most hacks don’t happen in one hour or even one day, it is a fact that once hackers gain control of the system they spend some time in navigating between servers and data to ensure that they get what they wanted. Leading RDBMS like Oracle, SQL & DB2 have rich auditing features that we can even track whether a particular column is queried from certain tables. For example if there is a credit card number column we can audit for any access outside of the application and immediately alert the entire system about it, which may help in avoiding the hacking activity to continue for all the sensitive data.

The below are some notes from Fine grained Auditing in Sql Server. The example creates a database audit specification called Audit_Pay_Tables that audits SELECT and INSERT statements by the dbo user, for the Human Resources.

```
CREATE DATABASE AUDIT SPECIFICATION Audit_Pay_Tables
FOR SERVER AUDIT Payrole_Security_Audit
ADD (SELECT, INSERT
ON Human Resources.Employee Pay History BY dbo )
WITH (STATE = ON) ;
```

Similar features are available in Oracle and DB2 databases also.

RDBMS Are Protected Against Human Errors:

If we have noticed the hack story, there was also threat to delete the data completely after is stolen. This is even a worse situation because not only the data is compromised but it is no longer available also. While the backup policies can help in this, it would be still better if the data analysts can go back in time after a hack & delete has happened to figure out the real state as it exists before the hack has happened.

Leading RDBMS like Oracle has got a concept of flashback database that let to view the past states of database or to return database objects to a previous state. SQL Server too has a snapshot concept which is more or less provide this functionality.

This feature cannot be implemented in a file system which does not support transactional integrity.

Summary:

While there is some merit in terms of scalability, performance and TCO perspective in tempting the enterprises to utilize file system based storage, however it has to be considered from case by case basis and any mission critical data is highly secured when gets stored in a relational database with proper design principles behind them. After all the cost of a hack and subsequent loss of face is much bigger than the licensing costs these proven RDBMS may incur.

Author: Srinivasan Sundara Rajan, GAVS Technologies.