

White Paper:

The Achilles Heel of your Data Security Strategy

Dynamic Data Obfuscation - the Next Frontier in Data Privacy

Common Knowledge

Ever since the inception of IT, a certain percentage of the IT Support Staff was granted access to the Production Environment for keeping systems up and running. As a natural by-product, Sensitive Private Information is constantly exposed to Privileged Production Support Personnel.

With today's Data Privacy issues, both practical and regulatory, it is important for enterprises to close this gap, especially those that make use of outsourced, cross-border resources.

As the Information Tsunami continues to grow, the challenge to protect sensitive data becomes increasingly difficult. Pressure comes from two sources:

a.) Business Risk - With or without legislation, the enterprise cannot afford to leak information into the wrong hands

b.) Regulatory Compliance - New legislation is appearing in constantly at all levels, State, Federal and Internationally. HIPAA, GBL, SOX, Patriot Act, SB1316, etc. are just the tip of the iceberg.

The Problem

Even with a solid Data Security Strategy that includes, Identity Management, Change Control, Access Controls, Database Access Monitoring, Encryption and End-Point Protection there is a gap that leaves Sensitive Private Information available to those who we only hope that we can trust. This paper introduces the concept of Source-Point Protection, the complement of End-Point Protection.

Source-Point Protection is simply security at the database layer. The concept of database encryption is not new, but the current methods all have one thing in common. By nature, the obfuscation is static. This is problematic from a number of perspectives. Static Data Obfuscation is processing-intensive, inflexible and most of all not practical for securing production data.

➤ Overhead

Virtually every data disguise solution is dependent on an ETL process where the real production data is copied and written to a destination with obfuscated values. Even using replication with special Views, enormous volumes are resources are consumed in the processing and storage of the encrypted databases.

The logo for DynamicDB, featuring the text "DynamicDB" in a stylized, italicized font with a red outline, set against a black background.

- Inflexible
Flexibility is sacrificed as there are different obfuscation requirements based on the various user communities that access the databases. When values are physically stored in the databases, it is not possible to obfuscate the same column of data with different rules. A simple example of this would be SSN (123-45-6789), where one group of users should only see the last four digits (xxx-xxx-6789), and another group should only see the first three digits (123-xxx-xxxx). In order to accomplish this, two different versions of the column must be maintained.
- Impractical
For obvious reasons, at the end of the ETL process, the source remains unchanged. However, this represents the Achilles Heel of the entire data security strategy. When people talk about Fort Knox, they do not talk about how thick the concrete walls are, they talk about who is watching the gold. The same applies here.

Privileged users, such as Production Support personnel, have the responsibility and the power to fix any problems that are a threat to data and application availability. They keep the business up and running. They also possess the Keys to the Kingdom. The database security model begins at the lowest possible level of access - Browse/Read-Only. Every level of privilege higher than that includes Browse.

- Outsourced Production DBA's. They have full authority to access any production database and perform radical brain surgery. However, even the brain surgeon doesn't have access to the patient's mind. Production Support people are regularly exposed to the secrets of the inner mind of the enterprise. CISO's, and even CEO's, lose sleep over this.
- QA staff. In order to create scripts, the QA Engineer logs onto the production system and runs through several transactions. Although they need to run the transactions, they should not be looking at sensitive data.
- Help Desk. In order to solve user problems, they are able to look at the screens with the users. They need to help the user, but they should not be looking at sensitive data.
- Training. In banks it is called "Model Bank"; insurance calls it "Model Office"; others call it "Parallel Production". These are Production-like environments with real accounts, real patients, real data that is used for training business professionals. However, there is TMI (Too Much Information) that is available to the wrong people.



Solution

There is a new paradigm in the concept of data masking. Gartner refers to it as Dynamic Data Masking. Targeted at the Production Environment, implemented at the SQL*Net Protocol Layer, an in-line proxy acts as a database listener. In-bound SQL is intercepted, inspected and tweaked just prior to execution. Established security policies can identify requests for sensitive data and apply an appropriate action.

The most frequently deployed action is to re-write the SQL statement before it is actually executed. An example would be to change `SELECT SSN` to `SELECT SUBSTR(SSN,1,3)//'xxxxxx'`. This tells the database to take SSN# 123-45-6789 and return it as 123-xx-xxxx. Another example would be to change `SELECT ENAME` to `SELECT SCRMBL(ENAME)`, which would take the name "Tiger" and change it to "Phil".

This concept allows for different users to see different iterations of the same data. For example, one set of users should see the first 3 digits of SSN, while another set of users should only see the last 4 digits of SSN.

One large telecom company has outsourced 50% of their Production DBAs who have the same Oracle privileges, do the same work, and use the same tools as the employee Production DBAs at the headquarters. With Dynamic Data Masking, the non-employee Production DBAs can still do their jobs, but when they browse the tables, the sensitive columns are masked; yet the employee Production DBAs see the unmasked data.

A large financial services company is using this solution to enforce the usage policies of their development tools. For example, many developers have CREATE privileges, but it is a policy that only DBAs should create tables. Through the use of the in-line proxy they are able to Block these requests and send a message back to the developer suggesting that they contact the DBA group.

This approach also provides a complete audit trail of database access activity enabling you to answer Who is accessing What, When, Where and How. Alerts and Reports are a natural by-product.

ActiveBase Security is the only commercially available technology on the market that does this. ActiveBase Security is available through Dynamic Database Solutions.

For more information please visit www.dynamic-db.com

