Craig S. Mullins

Return to Home Page

December 2007 / January 2008

zData Perspectives

by Craig S. Mullins

The Evolution of Database Security.

Protecting the data in our enterprise databases is extremely important. But what exactly does that mean? Oh, at one level we have the database authorization and roles built directly into the DBMS products. You know what I'm talking about: GRANT and REVOKE statements that can be used to authorize access to database objects, resources and statements. This level of database security is important, but is by no means sufficient. And many DBMS products are evolving to deliver additional security.

For example, DB2 offers multi-level security, which gives you the ability to protect data and authorize use of data at the row level. A multilevel security system allows the protection of data based on both traditional discretionary access controls, and controls that check the sensitivity of the data itself through mandatory access controls. These mandatory access controls are at the heart of a multilevel security environment, which prevents unauthorized users from accessing information at a classification they are not authorized to, or changing the classification of information they do have access to. These mandatory access controls provide a way to segregate users and their data from other users and their data regardless of the discretionary access they are given though access lists, etc. The primary arena where multilevel security is valuable is government agencies that need a security environment that keeps information classified and compartmentalized between users.

Another method to better secure database data is through encryption. But there are problems with encrypting database data. First of all, encryption is supported differently in every DBMS, so you need to dig in and understand how your particular DBMS supports encryption, and whether the support is sufficient or if you need to augment it with third party software. Also, performance is an issue. It takes CPU cycles to encrypt and decrypt that data, so is security more important than rapid access? Even more problematic is indexed access. If indexed columns are encrypted, the DBMS will sort the encrypted strings -- and they won't match the real, unencrypted data. So you cannot get indexed access paths for encrypted columns. Also keep in mind that some types of encryption require application code to be changed and nobody wants to do that, do they?

There are actually two types of encryption with respect to database data - encryption at rest and encryption over the wire. Basically, to this point, we've discussed encryption at rest -- that is, encrypting the database data on disk. But

there are also encryption products that will encrypt the data before it is sent across the network and decrypt it once it is received. Encryption over the wire is helpful to prevent surreptitious access to your data as it flies throughout your network, but it won't help combat thieves who target the database files on disk.

Another burgeoning field is database access auditing. This type of solution monitors database activity (INSERT, UPDATE, DELETE, and even SELECT) and reports on who is accessing and changing what data when. Such information can be very helpful to ensure that only appropriate personnel are accessing appropriate data within the database. Such solutions can help you to track the activity of privileged users (such as SYSADMs). And they can be invaluable in terms of being in compliance with regulations such as PCI-DSS and HIPAA.

Taking things yet another step further, auditing data can be achieved in an active manner - - let’s call it data trend monitoring. This type of solution monitors all database requests as they happen and discovers access patterns and trends. You can then configure policy-based access and control such that invalid data requests and/or modifications can be stopped and alerts can be generated when such actions are attempted. Trends can be monitored so that any behavior outside of the norm can be highlighted and investigated. This can be important, for example, to watch for suspect activity initiated by authorized users. Some analyst's suggest that such activity is more pervasive and potentially damaging to data than external attacks. And it can be very hard to detect.

All of these evolving database security features and products can be very useful to bolster existing database security and auditing tactics. So database security is evolving to enable better control over the precious information stored in enterprise databases. And like biological evolution, I would expect that we will see even

more types of database security offerings over time... although the timescale will most certainly be much shorter than the geological timescale of evolution.

From zJournal, Dec / Jan 2008

.

© 2008 Craig S. Mullins, All rights reserved.

Home.