November/December 2010

On Technical/ Security

EnterpriseOne® Security Best Practices, Part 2: Looking In *and* Outside the Box

By Gregg Larkin

E1 ***Editor's Note:*** *Tasked with security for JDE®? It's a never ending challenge for most administrators, with so many aspects of the system to consider and so many opportunities to miss something crucial. In part two of his series on EnterpriseOne security best practices, Gregg Larkin delves deeper into the system itself and presents some best practices with regard to web servers, integration servers, local browsers, and more.*

EnterpriseOne security is a big topic. I covered some of the best practices in part one of this series "*EnterpriseOne Security Best Practices: Looking Outside the Box*". In that article, I wrote about system-wide issues, database issues, and securing the configuration files on your enterprise and application servers. In this article, I'll cover the web servers, integration servers, tips for configuring the local browser, and some of the best practices for the EnterpriseOne.

Inside the Box – Application Security Best Practices

In this section, let's take a look under the hood and see how EnterpriseOne handles security and discuss best practices. This is not meant to be a replacement for a security class. If you are charged with defining and maintaining security, I would highly recommend the Oracle University security class. Think of this as a supplement from a security administrator with ten years of security under his belt (no wonder my pants are getting snug).

The Basics

Out of the box – JDE is not secured. This is done to allow for flexibility. In a small shop, roles will be fairly broad. As you scale up to a shop in a large, global corporation, jobs become more specialized and JDE roles become narrower. JDE is flexible enough to handle both extremes. Here is Security Golden Rule number one:

Out of the box – JDE is not secured.

Thou Shalt Implement an "All Doors Closed" Security Model

An "All Doors Closed" model starts with locking down all applications to all users, or in JDE – speak, lock down all applications to *PUBLIC. Then start to define your security roles by business function, granting back access to applications and reports as required. The security definition team is generally at least two or three people. Your team will consist of:

1. **Functional Experts:** People who know the application inside and out, and know how business people will use it. The functional experts will often work directly with the business users, gathering their functional requirements and then modeling the role. Usually they will start the modeling process by building a custom menu of applications and reports. Then they will refine the menu to fit work processes. Finally they will make judgment calls as to whether users will have full access, read access, or something in between.

2. **Developers:** A developer or two will also be involved to lend further expertise, create new applications, or new versions of applications and reports to fit the business requirements. Developers will also be involved in creating and modifying menus.

3. **CNC/Security Officer:** The CNC/Security officer's job is to create and maintain the security records, and to create and maintain the user accounts and roles.

This is an iterative cycle – the Functional expert does the definition. The Developers extend JDE to fit the business requirements; the CNC creates the users, roles, and security records. The CNC builds and deploys new packages to roll out the new applications and new and modified versions. Once a role is defined, the functional expert, often with help from the end users, tests out the new menu, adding and refining security over and over until it fits the requirements.

Defining JDE security is a tedious process. There are thousands of applications and reports to define. That takes us to Golden Rule number two:

Thou Shalt Not Blow Off Security Until The Last Phase of the Project

Security definition is the Rodney Dangerfield of JDE projects, it "gets no respect"-- until you've lived through a project and found out the hard way that defining security is a long, hard, tedious task. New users are prone to make mistakes. If they have too much access their mistakes are going to be harder to fix. If you define security well from the start, and you leave enough time in the project plan to define and test it, then you will not have to face lots of post go-live data cleanup. Wait until the end, give your users too much access, and you will be cleaning up data errors for years.

Tip: Consider Buying a Third Party Security Tool

There are two terrific tools out there on the market that can save hundreds of hours of time during the initial "from scratch" security definition phase. If you are charged with building security from scratch, check out these two vendors: Q Software and ALLOut Security. Both vendors have shortcut techniques to speed up the definition process. ALLOut Security has a utility that looks at a menu, then determines what applications and reports are on that menu. It then goes further and dives down two more levels and lists all of the applications called by the other applications. For example, any time you are in a grid and you see a little pop-up "help" window, that is a separate application. All of the row exits lead to separate applications, and so on. Those row exits need to be secured. ALLOut asks the question in a user friendly format, as to whether you want to grant full access, read access, or something in-between. Once it has all of the information, it creates the security role. Even ALLOut admits that it is not 100% fool-proof, but it gets you a long way toward defining role security. (disclaimer – I have worked with ALLOut's product, but have just seen demos of Q Software's product. I work for neither). From what I have seen of Q Software's product, they have similar functionality. Both vendors are easy to work with and are worth asking for a demonstration. Enough said…

So, what's the point, you ask? One way or the other, you will need to go through the process of identifying all of the applications and reports that the target users will require. You don't want to give users too much access; you just need to give them what they require to do their job. If you