JDEtips Journal January / February 2006 Volume VII Issue 1

JDEtips On Technical / Security

Passing a Sarbanes-Oxley Security Audit: Menu Security vs. Deny *All Security

By Michael Moorman

*Editor's Note: We all know that when our software offers us flexibility, it also forces us to make many decisions. Some must be based on practicality, some are cost-based, and some are purely technical. Now we are faced with an entire new level of decision-making based on regulatory requirements; most specifically in this case, the Sarbanes-Oxley requirements. Michael Moorman demonstrates the primary differences between Menu Security and Deny *All Security, and shows us how and why the latter is going to get you a lot farther with your SOX audit.*

Since the Sarbanes-Oxley Act of 2002 was passed in an effort to restore investor trust and confidence, all publicly traded companies now undergo a Sarbanes-Oxley (SOX) audit each year. Even many privately held companies are holding themselves to the same set of standards. The Act, as well as the new rules implemented by the SEC, introduced new requirements for public companies, corporate officers, audit committees, boards of directors, and accounting firms. Companies must now pass an external audit to be considered SOX compliant: whether your company will pass this audit with EnterpriseOne software installed will be determined by what security model you have implemented.

There are two main security models in EnterpriseOne: Menu Security and Deny *ALL objects. While this article highlights the advantages and disadvantages of both security models, as you will read, the deny *ALL model will pass the SOX audit. I will show you how to set up this model and also touch briefly on security external to the software and how it will affect the audit. The intent of the article is to help those individuals responsible for implementing an EnterpriseOne security model, therefore we won't discuss separation of duties or what each specific role within your organization should or should not have access to.

Disallowing fast path is a must in menu security.

Menu Security

The menu security model creates tasks and task views in Solution Explorer or menus in Enterprise One Explorer, and grants them to the different roles. Typically, most companies allot tasks for each division (Payroll, AP, AR, etc.) and only allow each division role see its own tasks/menus. Listed under each task/menu is the application the user in the role is able to run. By default, in the Security Workbench application (P00950), end users are able to run all applications. Therefore, your application access is secured at the task/menu level and not in Security Workbench. You can, however, take away certain applications or disable row and form exits in Security Workbench, depending on your company's security requirements.

Disallowing fast path is a must in menu security. By disabling fast path, users will not be able to open an application they are not allowed to run. Having fast path turned off, as well as having their menu/tasks assigned to them should, in theory, allow users to run only the applications they are allowed.

The biggest advantage to this security model is that it is quicker to set up than the deny *ALL security model. There is also not much security setup in the Security Workbench (P00950). And, you don't have to worry about applications that call other applications. For example, running the Batch Processing application (P0011) calls the General Ledger program (P0911). In the deny *ALL applications, you would have to grant access to both programs for the role to be able to run the Batch Processing application (P0011)

The disadvantages to menu security are row and form exits and creating shortcuts. Most applications have row and form exits. There is an exit bar in every application, as shown in Figure 1. In this secu-

rity model, you will have to search through all exit rows to make sure that a role doesn't have unauthorized access to an application.. This requires drilling down through each application's row and form exits, which is both a time consuming and error-prone effort. Here's an example of how easy it would be to miss something: when you open P0011, you can access P0911 from the Form exit. You would then have to search all the row and form exits in P0911. P0911 calls P01012. Once again, you would have to go through all the row and form exits for P01012. It's a vicious cycle in which it's easy to miss an application that should not be accessed by this user or role.

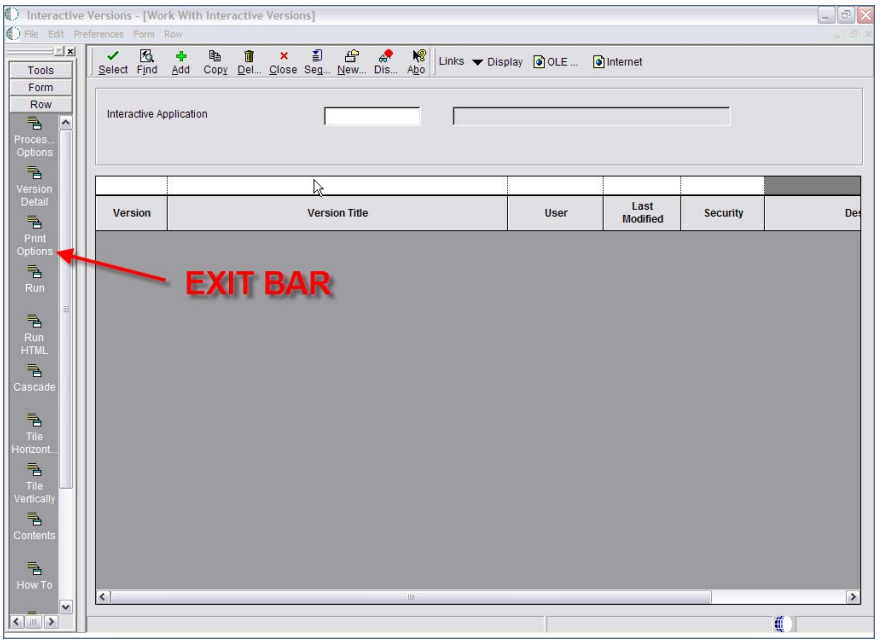

Figure 1 – Exit Bar

The other major disadvantage is creating shortcuts. The ability to create shortcuts is what leads most menu security models to fail the Sarbanes-Oxley audit. In Enterprise One, you can create a shortcut to an application. Once the shortcut is created, a user can email the shortcut to another co-worker in another department. As long as the co-worker has a valid User ID with EnterpriseOne, they can run the application from the shortcut. Let's say I have access to the Payroll Workbench application. I could create a shortcut to this application and email it to Bob in Accounting. Bob double clicks on the shortcut, and it will prompt him to sign in. Once he signs in, the application will open. Even though Bob doesn't have the Payroll application on his menu/task view, the system will still allow him to run it. Since menu security never denies applications, you're vulnerable to users running applications that they shouldn't be able to run.

Deny *ALL Security

As the name suggests, the deny *ALL security model is set up by denying all applications for all roles and granting back access to only the applications a user needs to run. A company with this model implemented correctly will pass the SOX audit. Just as with the menu security model, there are advantages and disadvantages to this model.

The biggest advantage is that users are only able to access those applications they are allowed to run. You don't have to worry about shortcuts being created or row and form exits, as found in the menu security model. You can allow fast path for all users, which is a major benefit to the end users. The deny*ALL security model is also easier to maintain once it is implemented.

A disadvantage to this model is the time it takes to implement,especially in the beginning. Because you have to take away all applications from all roles at the start, you need to understand what applications to grant back. You need to understand what applications and batch jobs (UBE) are called by other applications and batch jobs. And, of course, users don't like to be restricted in what applications they can access and will fight to be granted more, thus slowing implementation further. However, if you have a good security plan, coupled with management support, you can decrease the time it takes to implement this model.

Deny *ALL Setup

To set up the deny *ALL security model, you need to take away all applications from the end users and grant view-only for action security. To accomplish this, you will take away all applications from the *PUBLIC role, which applies to all users in the system. EnterpriseOne will check security at the user level first. If security is found for an application at the user level, it will be applied. If security is not found at the user level for the requested application, the system will check at the role level. If security is found for that role, it will be applied. If not, *PUBLIC security will apply for that user on that particular application.

To set up application security, open up Security Workbench (P00950).