JDEtips
JD Edwards Knowledge. Transferred. Sustained.

January 2015

On Technical/ Security

Security Best Practices

By Nathan Beaton, ALLOut Security

E1 Editor's Note: OK, everybody out of the pool. Now let's consider who really needs access to which functions and the best way to have a secure yet usable system. This article goes into the why and how to achieve this desirable result.

Ensure a Fully-Closed System

When implementing a deny-all security model, the best approach is to lock down at least both application and action code security. Because the default state of a JDE system is open in the absence of a security record, locking just one or the other results in a system that is potentially vulnerable.

Vulnerable Configurations:

*Only Application Closed for *ALL*

Assuming a configuration like the one shown below, the system appears to be secure on the surface in that no user may run an application unless explicitly granted the access.

Application Security		Action Security
Run	Install	No Record Defined
N	N	

The issue, however, comes when a user needs access to a specific object. The first part is easy – grant application access to that object; but we are left with two options in regards to action security.

1. Explicitly grant the appropriate level of action code security. While this may seem like the best choice, this approach requires two steps any time a user needs simple, inquiry-only access to an object. Also, this technique results in a potential increase in the number of security conflicts when multiple roles are granted to a user.
2. Leave action security alone because the user needs full access to the application. This causes an issue wherein it becomes difficult to tell if a user is supposed to have access to something or if the security administrator simply forgot to assign the appropriate action code security. Undertaking a project to move to a best-practice deny-all system results in difficulty analyzing appropriate access levels.

*Only Action Closed for *ALL*

Assuming a configuration like the one shown below, the system again appears to be secure on the surface in that no user may do anything within an application even though he or she may run that application.

Application Security	Action Security					
No Record Defined	Add	Change	Delete	OK	Copy	Scroll
	N	N	N	N	N	N

However, this configuration grants full authority for any user to run any and all update reports that are not explicitly denied access. Also, any application that does not rely on action-code security to prevent database updates is potentially vulnerable. For example, an application that uses a push button control or hyper exit to submit a database update would not be controlled by an action-only security model.

Menu Filtering

Menu filtering provides no security whatsoever from preventing a user from gaining access to an application via another method. A user may leverage a number of methods to subvert the supposed security imposed by menu filtering.

- Add a top-level folder to his/her favorites to obtain visibility into child folders.
- Use Batch Versions to launch any desired UBE.
- Use Interactive Versions to launch any desired interactive application.
- Craft a parameterized URL to use as the user's own personal Fast Path.

Essentially, use menu filtering to present a clean menu structure to your users, but don't get trapped into thinking that any access is prevented via this method.

A Better *ALL Approach

The recommended approach is to lock both application and action code security to prevent the issues described in the semi-closed models above.

Application Security		Action Security					
Run	Install	Add	Change	Delete	OK	Copy	Scroll
N	Y	N	N	N	Y	N	Y

This approach ensures that no user can access any application without explicitly defined access and ensures that inquiry-only access is the default when application security is granted. This eliminates the risk of a security officer forgetting to perform a function, thereby allowing too much access to the system.

Why Some Y's?

With most implementations of JDE now running via web clients, the install setting is less important, so ensuring that Run = N is sufficient to prevent access to an application.

The OK action security option should be set to Y to allow usable inquiry-only access to an application. With many Find/Browse or Search & Select forms, a user needs access to the Select button to drill down to a detail record or to select a value after a search.

When Action Security Doesn't Work and How to Fix It

Interestingly, action security does not always work as one would think, and it is critical that these intricacies be known to prevent unintended vulnerabilities in a system. Many of the action security settings control what buttons appear on typical Find/Browse form; i.e., Ok/Select, Add, Copy, and Delete all control what one