Shortcut properties are in the shortcut, so if they can read the shortcut, they can read the properties

devblogs.microsoft.com/oldnewthing/20120426-00

April 26, 2012

Raymond Chen

A customer wanted to know if "there was a way to hide the properties of a shortcut." We asked for an explanation of the problem they were trying to solve, so we could understand what their question meant. The customer liaison explained:

> The customer is insisting on this, even though I think it's really the wrong approach. They want to put a password into the parameters of a shortcut, but they don't want their employees to see the password when they right-click the shortcut and select Properties. We're trying to convince them of better ways of doing this, but right now they want to see if they can solve it by marking the field as "hidden" somehow.

If the password is anywhere in the shortcut file, the employees can dig it out. After all, the shell needs to dig it out, and since the shell runs with the user's privileges, in order for the shell to see it, the user must be able to see it. In other words, you can't hide anything in a shortcut because the user can just open the shortcut in Notepad and see all your "hidden" data. Or they can go to Task Manager and ask to see the command line. Or they can connect a debugger to Explorer and set a breakpoint on the `CreateProcess` function. It's like saying, "I want my employees to be able to bake cakes, but I don't want them to have access to an oven. To block access to the oven, I put a combination lock on the oven controls. On the other hand, I want to write a cake recipe that lets the employees bake cakes in the oven. Therefore, the recipe says *Step 5: Go to the oven and press 1234*. But now the employee can just read the recipe and find out the combination to the oven! Is there a way I can write a cake recipe that lets them bake a cake without revealing the oven combination?" The recipe executes with the privileges of the employee. If you want the employee to be able to bake a cake by following the recipe, then they need to be able to perform the steps in the recipe, and that means being able to go to the oven and press 1234. The oven analogy does provide some ideas on how you can solve the problem. For example, if you simply don't want employees to be able to email the oven combination to their friends with the subject line *Here's the combination to the oven!*, then change the way access to the oven is managed. Instead of putting a combination lock on the oven, put an employee ID card scanner on the oven that enables the oven controls if the employee has oven privileges. For the original problem, this means changing your database so that instead of using a single password to control access and trusting each client to use it wisely, it uses the security identity of the client to control access. (I'm assuming that the password on the command line is a database password.) On the other hand, if your goal is to prevent employees from using the oven to do anything other than *bake at 350°F for one hour*, you can change the employee ID card scanner so that it checks the employee for cake-baking privileges, and if so, sets the oven to bake for 350°F for one hour and locks the controls (except perhaps for a *cancel* button). If you have multiple recipes—say, cakes and cookies—the ID card scanner checks which recipes the employees are allowed to use and lets them choose which preset they want to activate. For the original problem, this means changing your database so that the user identity selects which operations are permitted on the database. Some users have permission to see only records for active clients, whereas others have permission to see all records, and still others have modify permission. If your goal is to prevent employees from doing anything other than *baking cakes according to this specific recipe*, then you need to move even more actions "behind the counter", because you have no way of knowing that "that pan full of batter" was created according to your walnut cake recipe, or whether it's some unauthorized recipe with extra cinnamon. If you don't trust your employees to follow recipes, then you need to take the recipe out of their hands. The instructions are now *Step 1: Order a walnut cake from the cafeteria*. For the original problem, this means changing your database so that instead of letting the employee access records directly, the employee submits the action to the database ("change client 23 address to 123 Main Street"), and the database verifies that the employee has "change a client address" permission, and if so, performs the record update.

Of course, if you want to ignore all the security guidance and "just hide the password in the shortcut file, why won't you answer my question?", you can just put the password somewhere other than the shortcut file. You could, say, have the shortcut run a batch file, and the batch file has the password.

Raymond Chen

Follow