We know it's insecure, but we want to do it anyway

devblogs.microsoft.com/oldnewthing/20060825-17

August 25, 2006

Raymond Chen

I remember a question from somebody who asked, paraphrasing:

> We're writing a secure screen saver that the user can interact with. We're going to present the user with various types of information, and if they click on a hot link, we want to launch a web page on their desktop once the user unlocks the workstation. We know it's insecure, but we want to do it anyway.

Apparently these people didn't get the memo on security. Windows tries to make it hard for you to do this to a secure screen saver. The restrictions on secure screen savers have gotten tighter over time. Originally, secure screen savers were isolated by putting them on a separate desktop. Nowadays, Windows also runs the secure screen saver in a job object, and when the user unlocks the workstation, all processes in the job are forcibly terminated. Even if you came up with some sort of workaround for this, it's entirely possible that your workaround will be treated as a security hole and rendered ineffective in a future version of Windows. Suppose you find yourself some workaround and are willing to concede that your technique is living on borrowed time. It's still a bad idea. One of the aspects of security that doesn't get much attention is repudiation. Responding to the user's actions from a secure screen saver to do anything other than unlock the workstation gives the user plausible deniability. "Yes, I know it's on your auditing logs, but I assure you, I didn't click on that link. When I came back from the printer room and unlocked my workstation, this web site appeared. It must have been somebody who wandered into my office." If your screen saver does anything nontrivial, it becomes something the user can plausibly deny, because any random person walking by could have done it. You have an untrackable and unattributable action. Network security administrators really get the heebie-jeebies when you say "untrackable and unattributable action". I'm told that when people ask for this sort of "interactive secure screen saver", they typically have some sort of process control program that they want always to be available. The thing is, if you're going to trust random passers-by with your control program, then you have basically decided that your computer's physical security is already assured. In that case, you may as well create a special account, configure the computer to auto-logon with that account, and put the control program in the special account's startup group. Just run the program like normal. Don't try to pretend that wrapping it inside a "secure screen saver" accomplishes anything.

Sidebar: While the rules for secure screen savers have gotten tighter over time, the rules for insecure screen savers have gotten more and more relaxed. Insecure screen savers are run on the user's desktop so that they can do the sorts of funny things that they got away with on Windows 95 such as taking a snapshot of the user's screen and using it as the basis for a jigsaw puzzle that it animated. Windows used to kill insecure screen savers as soon as the user touched a key or moved the mouse, but that behavior was disabled in order to allow people to write "interactive screen savers". (Remember PointCast?) Some people might argue that allowing insecure screen savers to interact with the user was a bad idea. (Yes, Internet Explorer at one point had a screen saver that did something like this. I'm told that in follow-up surveys, no customers actually admitted to liking the feature.)

Raymond Chen

Follow