You don’t optimize for the case where somebody is mis-using your system

devblogs.microsoft.com/oldnewthing/20070719-00

July 19, 2007

Raymond Chen

Commenter Rune Moberg asks why it takes the window manager so long to clean up 20,000 leaked objects.

Well, because you’re not supposed to leak 20,000 objects. Why optimize something that people aren’t supposed to be doing? That would just encourage them!

Actually, this scenario is doubly “don’t do that”. Creating 10,000 was already excessive; 20,000 is downright profligate. And in order to create 20,000 leaked objects in the first place, *you have to override the default limit of 10,000*. Surely this should be a clue that you’re taking the system beyond its design parameters: After all, you had to go in and change a system design parameter to get this far!

The window manager does clean them up eventually. Nothing goes wrong, but the experience is frustrating. Hopefully that’ll be a big clue that you’re doing something wrong.

Nitpicker’s corner

Of course, you have to apply the principle of “don’t optimize for abuse” with your brain still engaged. If the abuse can lead to a denial of service attack, then you have a security issue. (That’s not the case here, however. In order to leak 20,000 windows, an attacker has to be able to create 20,000 windows directly, rather than going through an intermediary such as a scripting engine, since the scripting engine will destroy the window rather than leak it. If an attacker can create windows directly, then why bother creating 20,000? Create one and use it to annoy the user.)

Raymond Chen

Follow