How come the technique for launching an unelevated process from an elevated process doesn't work?

devblogs.microsoft.com/oldnewthing/20150618-00

June 18, 2015

Raymond Chen

A customer was following the Execute in Explorer sample to launch an unelevated process from an elevated process. (A sample which I rehashed some time ago.) The customer reported that the resulting process was still elevated.

Upon closer inspection, the customer had disabled User Account Control (UAC).

If UAC is disabled, then the ability for an administrative user to launch an unelevated process no longer exists.

Since people like tables, here are some tables.

In the classical world without UAC, administrators are administrators, and standard users are standard users. In other words, processes run by administrators are always elevated, and processes run by standard users are always non-elevated.

UAC disabled		
	Process type	
User type	Elevated	Non-elevated
Administrator	•	
Standard		•

UAC added a new option to the table: The administrator who voluntarily relinquishes administrative privilege and runs a process non-elevated.

UAC enabled	
User type	**Process type**

	Elevated	Non-elevated
Administrator	●	★
Standard		●

In words: In the classic non-UAC world, an administrative user can run processes elevated, and a standard user can run processes un-elevated. If UAC is enabled, then a new combination becomes available: An administrative user can run a process non-elevated.

If you disable UAC, then you are back in the classic world, where there is no such thing as an administrative user running a non-elevated process. It's therefore no surprise that when you try to run the process unelevated, it still runs elevated.

You can look at this issue another way: If UAC is disabled, then Explorer runs elevated. And therefore, if you ask Explorer to run a process, that process runs elevated too.

It turns out that the customer turned off UAC because they didn't want to see any UAC prompts; they wanted their program to elevate silently, yet launch child processes unelevated. From a security-theoretical point of view, this is not an interesting configuration: If you allow silent elevation, then those child processes can just silently elevate themselves, and your attempt to run them unelevated accomplished nothing.

If you disable UAC, then the only way to get both elevated processes and unelevated processes is to run the elevated processes as one user (an administrator) and the unelevated processes as another user (a standard user).

Raymond Chen

Follow