What's the point of giving my unnamed object proper security attributes since unnamed objects aren't accessible outside the process anyway (or are they?)

devblogs.microsoft.com/oldnewthing/20150604-00

June 4, 2015

Raymond Chen

Recall that the NULL DACL grants total access to everybody. Both parts of this sentence are important to note.

- Everybody: This means everybody. No authentication required. It includes Guest. It includes Anonymous. It includes port scanners. It includes that creepy guy who hangs around the convenience store late at night.
- Total access: This is more than just read and write access. It includes taking ownership. It includes deleting the object. It includes changing the object's security to *lock you out of your own object*.

Suppose you're a bit lazy and you decide, "Eh, I know that a NULL DACL grants total access to everybody but since I'm creating an unnamed object, nobody can access the object anyway, so it really doesn't matter what security attributes I put on it, right?"

Um, no. Unnamed objects can be accessed. It's just that they can't be accessed by name.

Other ways of getting a handle to an unnamed object are to duplicate the handle (and the `DuplicateHandle` function will gladly duplicate handles across processes) or to inherit it.

Therefore, if you create an object with a NULL DACL, you are exposing it to everybody who has `PROCESS_DUP_HANDLE` permission. Which is a lot of people. And one of them may in turn be rather sloppy with their security and let the handle escape further.

Just do the right thing. Give the object the correct security attributes. (If you don't intend to use the object outside your process, then the default security attributes are probably just fine.)

Besides, it's easier to create an object with default security than it is to create an object with custom security, so just do the default thing and save yourself a bunch of typing.

Raymond Chen

Follow