How come my CreateMutex call sometimes fails with Access denied?

devblogs.microsoft.com/oldnewthing/20170116-00

January 16, 2017

Raymond Chen

A customer reported that they had a system where multiple processes share a mutex. Each process calls `CreateMutex` to create the mutex or obtain a handle to the existing one. But the customer found that sometimes, the call to `CreateMutex` fails, and `GetLastError` reports that the reason is `ERROR_ACCESS_DENIED`. What could cause that?

Specifically, the two processes are a UI process and a service. Both processes use `Create-Mutex` to create or access the mutex, passing `NULL` as the security attributes.

Okay, so issue is that the two processes are running under different identities with different privileges. Even though you think you are creating the mutex in both places with the same security attributes (because you're passing `NULL` both times), the effect of the `NULL` is different depending on who is calling.

What's probably happening is that most of the time, the mutex is created by the UI process first, so the mutex gets the access control list that grants access to the user under which the UI process is running, and to the SYSTEM account, The service is running under the SYSTEM account, so it gets access.

But once in a while, it's the service that creates the mutex first. In that case, the access control list on the mutex grants access only to the SYSTEM account, in which case the UI process cannot access the mutex.

The customer reported that they saw this behavior only on builds 14393 and higher and were wondering what could be causing it. My guess is that something is happening on build 14393 that wakes up the service sooner than it used to, causing the service to be the one to create the mutex, and that's what's preventing the UI process from gaining access.

In these sorts of cases, where two processes with different permissions need to share a securable object, the general principle is to have the high-permission process create the object and set the permission so that the low-permission process can access it. If you let the low-permission process be the one that creates the mutex, then you have a potential squatting attack.

Raymond Chen

Follow