What happens if you call RevertToSelf when not impersonating?

devblogs.microsoft.com/oldnewthing/20161102-00

November 2, 2016

Raymond Chen

A customer wanted to know what happens if you call `RevertToSelf` from a thread that is not impersonating. "Does the call succeed or fail? This particular scenario is not explicitly discussed in the documentation. We have a bunch of places in our code that say `if (impersonating) RevertToSelf();` and we were wondering whether the `if` test was really necessary."

The answer to the question is that calling `RevertToSelf` when the thread is not impersonating will return success without doing anything (because the thread is already not impersonating).

However, that doesn't mean that you can blindly remove all your `if` tests. You don't want to over-revert either. Consider:

```
// Error checking elided for expository purposes.
void DoSomething()
{
 bool impersonating = false;

 if (!ThreadIsAlreadyImpersonating() &&
     ImpersonationIsNeeded()) {
   StartImpersonating();
   impersonating = true;
 }

 DoWork();

 if (impersonating) {
  RevertToSelf();
 }
}
```

If you remove the `if (impersonating)` and unconditionally revert, then you have a security defect if the thread was already impersonating, because your modified code will unconditionally revert and prematurely end the existing impersonation.

So yes, it's okay to call `RevertToSelf` when the thread is not impersonating, but that doesn't relieve you of the responsiblity of knowing when to revert.

Raymond Chen

Follow