Why can’t you apply ACLs to registry values?

devblogs.microsoft.com/oldnewthing/20090123-00

January 23, 2009

Raymond Chen

Someone wondered why you can’t apply ACLs to individual registry values, only to the containing keys.

You already know enough to answer this question; you just have to put the pieces together.

In order for a kernel object to be ACL-able, you need to be able to create a handle to it, since it is the act of creating the handle that performs the access check.

Creating a handle to the value means that we would need a function like `RegOpenValue` and corresponding `RegQueryValueData` and `RegSetValueData` functions which take not a registry key handle but a registry value handle.

And then you’ve basically come full circle. You’ve reinvented the 16-bit registry, where data was stored only in the tips of the trees. Just change *value* to *subkey* and you’re back where you started.

What would be the point of adding an additional layer that just re-expresses what you had before, just in a more complicated way?

Commenter bcthanks wondered why we didn’t abandon values and just stored everything in subkeys, like the 16-bit registry did. Well, if you want to do that, then more power to you. Though it would make it difficult for you to store anything other than `REG_SZ` data in the registry. If you wrote a `REG_BINARY` blob to the default value of a subkey, what should be returned if somebody called `RegQueryValue` which always returns a string?

Raymond Chen

Follow