Why does the CREATOR_OWNER SID sometimes reset itself to the object's current owner rather than its original owner?

devblogs.microsoft.com/oldnewthing/20180613-00

June 13, 2018

Raymond Chen

I noted some time ago that the `CREATOR_OWNER` security identifier (SID) is not a shorthand that refers to the object's current owner. Rather, the `CREATOR_OWNER` SID is a template that is applied when the object is created. When the template is applied, all occurrences of `CREATOR_OWNER` are replaced with the object's current owner, and it is the replaced SID that controls access. Changing the object's owner doesn't cause these access control entries to be recalculated;[1] they continue to refer to the captured value.

A customer observed this phenomenon when they created a folder with an inheritable access control entry (ACE) for `CREATOR_OWNER`. They observed that those access control entries were indeed propagated to child objects, with the `CREATOR_OWNER` changed to the object's actual owner. Furthermore, if they went to the Security properties and changed the child object's owner, the ACE was not recalculated to update the ACE's SID from the old owner to the new owner.

However (and this is the weird part), if they use the Security properties to make some unrelated change to the object's access control list (ACL), then this has a side effect of recalculating the ACEs and updating the `CREATOR_OWNER`-sourced ACEs to refer to the new owner.

This recalculation is not being done by the security infrastructure. It's being done by the ACL editor.

When you change the access control list for an item, the ACL editor calls `TreeSetNamedSecurityInfo` and passes an ACL that consists only of the non-inherited ACEs, and it sets the `UNPROTECTED_DACL_SECURITY_INFORMATION` flag, which means "Oh, and also inherit ACEs from my parent, as if I were newly-created."

In other words, the ACL edit deletes all the ACEs that were obtained by inherance, and then creates new ACEs based on the current parent's inheritable ACEs.

The ACL editor is trying to be helpful, but it ends up being confusing.

[1] What would this recalculation even mean if the object was moved to a new folder after being created, or if the containing folder's access control list were modified in the interim? I guess you could have a bit somewhere in the ACE that says, "This was originally created from a template that used `CREATOR_OWNER`." The closest thing to that is the `INHERITED_ACE` bit, which says "This ACE was autogenerated via inheritance," but it doesn't give any information as to what the original ACE was. Suppose the object's current owner is Bob. If an ACE applies to Bob and has the `INHERITED_ACE` bit set, it could mean that the original template ACE's SID was `CREATOR_OWNER` that was changed to Bob during template application because Bob was the original owner, or it could mean that the original template ACE's SID was `CREATOR_GROUP` that was changed to Bob during template application because Bob was the original group, or it could mean that the original template ACE's SID was Bob all along.

Raymond Chen

Follow