How do I create a directory where people can create subdirectories but cannot mess with those created by other users?

devblogs.microsoft.com/oldnewthing/20160524-00

May 24, 2016

Raymond Chen

A customer was having trouble setting up security for a new file share for which they wanted a particular usage model:

> What we would like is for everybody to be able to create new files and folders on the share but not overwrite existing content. We can't find the correct set of ACLs that give us what we want. We found "Create Folders / Append Data", which sounds like it lets you create folders and append to existing files on the share, but it doesn't let you create files. Is that correct? What's more, when we tried it, it didn't seem to do what it says on the tin. We can create folders, and we can create empty files, but we are unable to write any content into those files. Maybe the permissions we are setting up don't make sense? Can you suggest a security configuration that gives us what we want?

"Create Folders / Append Data" access is one of the many two-headed permissions in file system security. It means "If applied to a folder, you can create folders. If applied to a file, you can append data." That's what the slash is trying to tell you.

What you can do is set "Create Folders / Append Data" on the root folder, but mark it is non-inheritable. This means that users can create folders in the root, but since it doesn't inherit, they will not be able to create folders inside any folders they create, nor will they be able to append to any files they create. (On the other hand, that seems to result in the situation described above, where you can create a file but you can't write anything to it, because writing to an empty file is equivalent to appending.)

But let's suppose that the customer's "not overwrite existing content" really means "not overwrite existing content created by other users". In other words, we want to let users create new content, and they can do whatever they want to the content they created, but they can't mess with content created by others.

As noted by my colleague Pavel Lebedinsky some time ago, the Windows temp directory is set up very similar to what you want.

```
C:\Windows>cacls temp
                BUILTIN\Users:(CI)(special access:)
                                   SYNCHRONIZE
                                   FILE_WRITE_DATA
                                   FILE_APPEND_DATA
                                   FILE_EXECUTE

                BUILTIN\Administrators:F
                BUILTIN\Administrators:(OI)(CI)(IO)F
                NT AUTHORITY\SYSTEM:F
                NT AUTHORITY\SYSTEM:(OI)(CI)(IO)F
                CREATOR OWNER:(OI)(CI)(IO)F
```

Administrators and SYSTEM get full access to everything, but that's not unusual. The interesting ACEs are the ones for Users and CREATOR OWNER.

The dual-headed-ness of many file system access mask values makes the output a little harder to understand, because `FILE_WRITE_DATA` means different things depending on whether you apply it to a file or to a folder. Let's see what we can figure out.

The Users ACE is marked Container Inherit (CI) which means that it applies to this folder and subfolders, but not to files. `FILE_WRITE_DATA` applied to a folder means `FILE_ADD_FILE`, so users can create files in this folder or any subfolders. `FILE_APPEND_DATA` applied to a folder means `FILE_ADD_SUBDIRECTORY`, so users can create subdirectories in this folder or any subfolders. And `FILE_EXECUTE` applied to a folder means `FILE_TRAVERSE`, so users can traverse through this folder any subfolders.

Recall that CREATOR OWNER is a placeholder ACE. Nobody actually has CREATOR OWNER; rather, it is a template that gets applied to newly-created objects. Therefore, the CREATOR OWNER ACE gets applied to files and folders that users create, and the user who created the file or folder is inserted as the owner. Here, the ACE is saying that users have full control over any folders or files that they create.

The security settings for the Windows temp directory are a good start for fiddling around with usage patterns like the one the customer is looking for. In fact, it may already be what the customer wants.

The customer thanked us for the information.

Raymond Chen

Follow