Anybody can make up a generic mapping

devblogs.microsoft.com/oldnewthing/20080929-00

September 29, 2008

Raymond Chen

Each component that uses ACLs to control access has its own idea of what `GENERIC_READ`, `GENERIC_WRITE`, and `GENERIC_EXECUTE` mean. It's not like there's a master list that somebody can make that lists them all, because I can make up a new one right here. Watch me:

```
#define GIZMO_QUERY_STATUS   0x0001
#define GIZMO_QUERY_MEMBERS  0x0002
#define GIZMO_START          0x0004
#define GIZMO_STOP           0x0008
#define GIZMO_ADD_CLIENT     0x0010
#define GIZMO_REMOVE_CLIENT  0x0020


#define GIZMO_GENERIC_READ  (STANDARD_RIGHTS_READ | \
                             GIZMO_QUERY_STATUS | \
                             GIZMO_QUERY_MEMBERS)
#define GIZMO_GENERIC_READ  (STANDARD_RIGHTS_READ | GIZMO_QUERY_STATUS)
#define GIZMO_GENERIC_WRITE (STANDARD_RIGHTS_WRITE | \
                             GIZMO_ADD_CLIENT | \
                             GIZMO_REMOVE_CLIENT)
#define GIZMO_GENERIC_EXECUTE (STANDARD_RIGHTS_EXECUTE | \
                             GIZMO_START | \
                             GIZMO_STOP)
#define GIZMO_ALL_ACCESS    (STANDARD_RIGHTS_REQUIRED | \
                             GIZMO_QUERY_STATUS | \
                             GIZMO_QUERY_MEMBERS | \
                             GIZMO_START | \
                             GIZMO_STOP | \
                             GIZMO_ADD_CLIENT   | \
                             GIZMO_REMOVE_CLIENT)


GENERIC_MAPPING GizmoGenericMapping = {
    GIZMO_GENERIC_READ,
    GIZMO_GENERIC_WRITE,
    GIZMO_GENERIC_EXECUTE,
    GIZMO_ALL_ACCESS,
};
```

It's not just kernel objects that use ACLs. Anybody who wants to set up permissions can use ACLs to control access. For example, the file server service uses ACLs to control which users can create new file shares, which users can view printer properties, which users can connect to administrative shares, all that stuff. There is no kernel object that these access masks apply to; they merely control who can do what with the service.

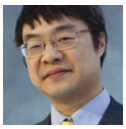
In that example above, a "gizmo" might be some sort of chat room with a member list. Some users may have permission to add and remove other members from the chat room; others have permission to open the chat room or shut it down. When a client wants to perform an operation on the chat room, the program obtains the security descriptor for the chat room and calls `AccessCheck` to see whether the caller has permission to perform the operation.

This is a totally artificial example. My point is that anybody can make up access bits and use them to control access to some sort of shared resource. That shared resource might be something you think of as a "real object" like a file or a process, but it could be some sort of

purely virtual construction like a chat room or a file share. Even if some sort of "complete list" were developed, anybody working in a basement can add a new one, and then your complete list is incomplete.

Bonus chatter: One of my colleagues points out that the mandatory integrity mechanism does have implications for generic mappings. I don't even understand that sentence, but there it is for you to ponder.

[Raymond is currently away; this message was pre-recorded.]



Raymond Chen

Follow