When people ask for security holes as features: World-writable files

devblogs.microsoft.com/oldnewthing/20041122-00

November 22, 2004

Raymond Chen

If I had a nickel each time somebody asked for a feature that was a security hole...

I'd have a lot of nickels.

For example, "I want a file that all users can write to. My program will use it as a common database of goodies."

This is a security hole. For a start, there's an obvious denial of service attack by having a user open the file in exclusive mode and never letting go. There's also a data tampering attack, where the user opens the file and write zeros all over it or merely alter the data in subtle ways. Your music index suddenly lost all its Britney Spears songs. (Then again, maybe that's a good thing. Sneakier would be to edit the index so that when somebody tries to play a Britney Spears song, they get Madonna instead.) [Minor typo fixed. 10am]

A colleague from the security team pointed out another problem with this design: Disk quotas. Whoever created the file is charged for the disk space consumed by that file, even if most of the entries in the file belong to someone else. If you create the file in your Setup program, then it will most likely be owned by an administrator. Administrators are exempt from quotas, which means that everybody can party their data into the file for free! (Use alternate data streams so you can store your data there without affecting normal users of the file.) And if the file is on the system partition (which it probably is), then users can try to fill up all the available disk space and crash the system.

If you have a shared resource that you want to let people mess with, one way to do this is with a service. Users do not access the resource directly but rather go through the service. The service decides what the user is allowed to do with the resource. Maybe some users are permitted only to increment the "number of times played" counter, while others are allowed to edit the song titles. If a user is hogging the resource, the server might refuse connections for a while from that user.

A file doesn't give you this degree of control over what people can do with it. If you grant write permission to a user, then that user can write to **any** part of the file. The user can open the file in exclusive mode and prevent anybody else from accessing it. The user can put fake data in the file in an attempt to confuse the other users on the machine.

In other words, the user can make a change to the system that impacts how other users can use the system. This sort of "impact other users" behavior is something that is reserved for administrators. An unprivileged user should be allowed only to mess up his own life; he shouldn't be allowed to mess up other users' lives.

Armed with this information, perhaps now you can answer this question posted to comp.os.ms-windows.programmer a few months ago.

Raymond Chen

Follow