Dubious security vulnerability: Granting access to SIDs that don't exist yet

devblogs.microsoft.com/oldnewthing/20230106-00

January 6, 2023

Raymond Chen

A security vulnerability report arrived that went like this:

> A user can gain access to arbitrary objects on the domain as follows:
>
> 1. Gain administrator access to the domain.
> 2. Modify the access control lists (ACLs) on objects of choice so that they grant permission to a security identifier (SID) that has not been assigned.
> 3. Go to a machine on the domain and generate SIDs (say, by adding new machine accounts) until one of them matches the SID that you planted in step 2.
> 4. Use that maching account to regain access to the objects.
>
> The system should not allow ACLs to contain SIDs that do not correspond to valid identities.

Okay, just to get it out of the way: If the first step of your attack is "Gain administrator access to the domain", you not only on the other side of the airtight hatchway, you've escaped the ship entirely and made it to the flagship vessel and gained control of its command center! There's no elevation of privilege: Once you are the domain administrator, you have control over all the computers in the domain.

No, what this is really reporting is that once someone compromises your domain, they can create backdoors that will let them back in. But that is also not interesting. Once somebody compromises your domain, you've already lost.

But let's look at the finder's concern about ACLs which contain SIDs that do not correspond to valid identities. This is actually a feature, not a bug.

Imagine you are setting up a card reader to control access to a secure building, and you program the card reader so that it recognizes employee ID cards, and you set a list of ID numbers for the employees who are allowed in. There's nothing to stop you from putting a fictitious employee ID number on the list, a number that does not correspond to any employee that has yet been hired. Of course, it also means that if that employee ID number

gets assigned to a new employee, and they try to enter the building, they will get through, because their employee ID number is on the list. (Pro tip: Do not reuse employee ID numbers.)

The card reader's job is to check the employee ID number, and if the number is on the list, it unlocks the door. If you didn't want to take the chance of a random employee being assigned your fictitious employee ID number, you shouldn't have put it on the list. But the card reader doesn't care that some of the numbers don't correspond to any known employee. The card reader doesn't even know which employee ID numbers are valid!

Okay, so let's go back to SIDs.

First of all, it would be a terrible performance penalty to have to validate every SID in every ACL and reject invalid ones. Determining whether a SID is valid might require a network call to the domain controller. If the machine doesn't have network connectivity, would you just reject all ACL changes?

Furthermore, even with network access, it may not even be possible to validate a SID. Suppose you get a SID for S-1-5-21-1004336348-1177238915-682003330-1001. Is that valid? To find out, you need to contact the domain controller for S-1-5-21-1004336348-1177238915-682003330. Do you even know who that is?

It's legal to add SIDs that don't exist, because they might exist in the future. For example, you might be on a Window Vista machine and set the ACLs on a file to grant access to S-1-5-32-578, which is the group of Hyper-V administrators. However, Windows Vista doesn't know about that group, which wasn't added until Windows 7. You are granting access in anticipation of the group existing: Once the system upgrades to Windows 7, that SID will magically begin to exist, and the file becomes accessible to Hyper-V administrators.

It's legal to add SIDs that don't exist, because they might exist in a place you haven't learned about yet. For example, you can grant access to the SID S-1-5-2-x1-x2-x3-x4-x5-x6-x7, which we learned earlier is an app package SID. The app may not be installed right now, but the user might install the app later, and you want the file to be accessible to that app. Or you can grant access to a SID that corresponds to a Windows Live ID[1] so that when that user adds an account to your computer, they get access to the file that you want to share with them.

If you connect to a network file server, you may want to update the access control list for a file on the server to grant access to a SID that you know about, but the server doesn't.

And what if a user gets deleted? Do all the existing ACLs suddenly become invalid? Does the system crawl every ACL on every hard drive (oh no, what about removable hard drives?) and forcibly delete SID entries for that user?

The security system doesn't try to validate the SIDs that you put in your access control list. If they don't correspond to any valid user, then no valid user will ever present that SID, and the entry is just junk DNA, sitting around taking up space but not doing anything. But once a user with that SID shows up, access will be granted. Because you told it to let them in.

[1] The Windows Live ID Sign-In Assistant allows you to generate SIDs from Windows Live accounts, so you can add them to access control lists (such as for files), so that when that account signs into the computer in the future, they will have access to the files that you shared with them. Somehow, Slashdot got the idea that this meant that Windows was gaining some sort of computer-to-computer file sharing. It's just a SID provider, so that you can set the security attributes on a file to say "Let billg@contoso.com access this file" (should he ever log onto this computer). It doesn't transmit the file anywhere.