How do I get the name of a SID, and what does it mean when the conversion fails?

devblogs.microsoft.com/oldnewthing/20240610-00

June 10, 2024

Raymond Chen

A customer had a file share, and they couldn't figure out who has access to some of the files. They pulled the access control list (ACL), extracted the security IDs (SIDs), converted them from binary to string form, and the result was `S-1-5-21-2127521184-1604012920-1887927527-72713`. "Why didn't the conversion work?"

This is a rather confusing question, because the conversion did work. You have the answer: It's `S-1-5-21-2127521184-1604012920-1887927527-72713`.

After some back-and-forth we learned that what the customer really meant was "How can we take this SID and convert it to something semantically meaningful to humans, like a person's name?"[1]

Programmatically, you can use `LsaLookupSids2` to ask the local system to find a friendly name for a SID. It will consult its local account database, the domain's account database, and maybe some other stuff. But sometimes it just shrugs its shoulders and says, "Sorry, I can't come up with a better name for this one."

The customer was confused by the possibility of a SID with no known name. How can the security system work if it doesn't know who a SID represents?

The security system doesn't care *who* a SID represents. When somebody tries to access a resource, it looks for matches between that person's SIDs and the SIDs in the ACL, and follows the instructions associated with those matches. For example, there may be an access control entry (ACE) that says "Allow `S-1-5-21-2127521184-1604012920-1887927527-72713` to have read-only access." The security system doesn't know or care who `S-1-5-21-2127521184-1604012920-1887927527-72713` is.

By analogy, suppose your IT department locks down your phone by pushing a list of phone numbers from whom it will accept calls. The phone doesn't know the names of the callers, but it doesn't need to know. It just takes every incoming phone call and sees if the number is on the "Allow" list. (This also avoids problems if somebody calls who happens to have the

same name as a person on your allow list. Since the phone number doesn't match, the call is not let through.) As a courtesy, your phone tries to find a name for the incoming call by looking in the phone's contacts or maybe by contacting a service. But that is best effort, and sometimes you just get "Unknown caller". Your phone doesn't know who they are, but the phone doesn't need to know who they are in order to make a block/allow decision.

You don't know who this 555-1212 number is, but it's on your Allow list. And you don't know who this `S-1-5-21-2127521184-1604012920-1887927527-72713` SID is, but it's there on the file's Allow list.

The customer said that they were using the Advanced Security Settings property sheet to look up the SIDs. That property sheet already uses `LsaLookupSids2` to look up friendly names for every SID, if known. If something shows up in `S-...` format, then it means that the system couldn't find a friendly name.

At this point, you need to follow the money backward. If you want to know who this 555-1212 number is, you can ask your IT administrator, since they are the ones who added that number to the Allow list. If you want to know who this `S-1-5-21-2127521184-1604012920-1887927527-72713` SID is, you can ask the file share owner or the file owner who it is. Presumably whoever put it on the access control list knows who it is. Otherwise, why would they have added it?

[1] It's not clear to me how they expected the manual algorithm to produce something semantically meaningful, since the manual algorithm consists of counting things and inserting dashes. Nowhere in the algorithm does a number like "72713" turn into a name like "Chris".