About Security Identifiers (SIDs)

First published: 02-Jul-2010

Introduction

Most IT professionals can recognise a Security Identifier (SID), such as S-1-5-32-544, but where do SIDs come from, what do they represent, what are they used for, how are they stored, and how do you translate and interpret them?

This document sets out to answer these questions.

What does a SID represent?

SIDs are random, unique identifiers that are used on Microsoft systems to represent security objects, such as user accounts, security groups, domains and computers. A SID is generated by the Operating System when an object is defined and assigned to the object for its entire lifetime.

SIDs should not be confused with GUIDs (Globally Unique Identifiers). Although SIDs and GUIDs serve very similar purposes, it is important to note that they also differ in many respects, particularly in their format, composition and application. SIDs perform a very similar function to UIDs and GIDs on UNIX systems.

While security objects also have user-friendly display names such as Administrator and Guest, it is important to remember that the Operating System and its parts reference an object by its SID and not by its friendly name. An object's SID never changes, so its SID remains the same even if the object is renamed or moved to another location.

An exception to this rule is when a security object is moved to another *domain*, because the SID's sub-authority components must change, although the object's previous SID history is typically retained. *However, this situation is outside the scope of this document.*

What are SIDs used for?

A SID is used by the Operating System to identify a security object. Some of the more common applications are in NTFS (NT File System), the System Registry, SAM (Security Account Manager) and Active Directory.

NTFS uses SIDs to link access and auditing rules for files and directories to their owners and trustees. In summary, SIDs are defined in Access Control Entries (ACEs) which are used to build Discretionary Access Control Lists (DACLs) and Security Access Control Lists (SACLs).

Similarly, in the System Registry, SIDs identify security objects and serve as a link to many properties stored in the Registry that relate to, and describe the object.

SAM uses SIDs to define and identify group members and as the link to an object's security properties, such as its Primary Group Identifier (PID), last password change date and status. In Active Directory an object's SID serves as the link between a domain object's SAM record and other properties for the object that are stored in the Active Directory database.

How are SIDs stored and represented?

Like all other computer data, SIDs are stored internally in binary format, although they are usually displayed in a more user-friendly format using the Security Descriptor Definition Language (SDDL) notation.

SIDs are variable in length, although a typical SID for an Active Directory domain object (e.g. CN=Administrator,CN=Users,DC=Research,DC=SekChek,DC=com) occupies 28 bytes.

For example, the SID for the domain administrator account may be represented in SDDL format as:

S-1-5-21-4279025473-3018771506-1539134433-500

Internally, the SID occupies 28 bytes and is stored in binary as:

00000001 - 00000101 - 00000000 00000000 00000000 00000000 00000000 00000101 - 00010101 00000000 00000000 00000000 - 01000001 10111111 00001100 11111111 - 00110010 11001100 11101110 10110011 - 11100001 01010011 10111101 01011011 - 11110100 00000001 00000000 00000000

which can also be represented as 28 hexadecimal character pairs:

01 - 05 - 00 00 00 00 00 05 - 15 00 00 00 - 41 BF 0C FF - 32 CC EE B3 - E1 53 BD 5B - F4 01 00 00

Note that separator characters ('-') have been artificially included in the binary and hexadecimal strings to highlight the various components of the SID and to make the values easier to read.

But, how is the SDDL format of the SID derived from its binary and hexadecimal equivalent? And what do the components of the SID represent?

First published: 02-Jul-2010

How do you convert a SID from its binary / hexadecimal format to SDDL?

The following diagram illustrates the components of a SID and the related text explains how to convert a SID to SDDL format from its raw binary or hexadecimal equivalent.

Components of a SID (S-R-I-S-S)

Note that the hexadecimal format of the above SID (S-1-5-21-4279025473-3018771506-1539134433-500) is shown on the first line of the table and the SDDL equivalent on the second.

SID prefix	SID revision level	The number of sub-authorities in the SID	The identifier authority value	Sub-authority 1 (little-endian)	Sub-authority 2 (little-endian)	Sub-authority 3 (little-endian)	Sub-authority 4 (little-endian)	Sub-authority 5 (RID) (little-endian)
	01	05	00 00 00 00 00 05	15 00 00 00	41 BF 0C FF	32 CC EE B3	E1 53 BD 5B	F4 01 00 00
S	1		5	21	4279025473	3018771506	1539134433	500

The SDDL representation of a SID can be summarised as S-R-I-S-S ('S' prefix, Revision, Identifier authority, Sub-authority, Sub-authority). For example:

SID prefix. The 'S' prefix indicates that what follows is a SID.

Revision level. The '1' indicates the version of the SID specification.

The number of sub-authorities in the SID. This is represented by '5' in the example above. *Note that the number of sub-authorities in a SID can vary and the value is not included in the SDDL representation of a SID.*

The identifier authority value. The value '5' represents the 'NT Authority'. The most common identifier authorities are:

0 Null Authority

1 World Authority

2 Local Authority

3 Creator Authority

4 Non-unique Authority

5 NT Authority

Sub-authority 1. '21' indicates that the remaining sub-authorities identify the domain. Note that the remaining sub-authorities can also be used to identify a computer.

Sub-authorities 2 to 4. 4279025473-3018771506-1539134433 identify the domain.

Sub-authority 5. The object's relative identifier (RID). '500' is the RID for the Builtin Administrator account. Note that any user-defined object (i.e. objects not shipped by Microsoft) will have a RID of 1000 or greater.

You probably noticed that it is easy to convert the 'S-R-I' parts of the SID, but how do you convert the sub-authority values from hexadecimal to decimal?

For example, Hex 15 00 00 00 that appears in Sub-authority 1 converts to decimal 352321536 and not to 21.

Well, a SID's sub-authority values are stored internally in little-endian byte order, so you need to reverse the order of the hexadecimal character pairs before converting them to decimal. So, when Hex 15 00 00 00 is reversed it becomes Hex 00 00 00 15, which converts to decimal 21.

Why do I need to know how to read or convert a SID?

There are various situations where this can be useful. One example is reading a memory dump, because the SID will only be displayed in its raw format, so if you don't know how to interpret a SID you won't be able to make any sense out of it.

Also, when a security object is deleted, Windows cannot resolve its orphaned SID to its friendly name, such as 'MyDomain\MyAccount' because the account no longer exists in the system's SAM database. A similar condition occurs if the system that holds the SID is permanently or temporarily unavailable.

In these situations, the object's SID, and not its friendly name, will be displayed when you view NTFS permissions on a file or directory or list a group's members.

These are just some examples of situations where it is useful to know which domain or machine the SID belonged to, so you can decide whether it is safe to delete the SID from the file's DACL or remove the SID from a group's member list.

About Security Identifiers (SIDs)

First published: 02-Jul-2010

Well-Known SIDs

Microsoft ships several well-known SIDs with their systems. These special, generic SIDs are the same across all systems and they cannot be changed. An example is the well-known SID S-1-5-32-544, which always represents the Builtin Administrators group.

Some examples of other well-known SIDs (extracted from MSDN) are:

S-1-5-32-544: Administrators

> A built-in group. After the initial installation of the operating system, the only member of the group is the Administrator account. When a computer joins a domain, the Domain Admins group is added to the Administrators group. When a server becomes a domain controller, the Enterprise Admins group also is added to the Administrators group.

S-1-5-32-545: Users

> A built-in group. After the initial installation of the operating system, the only member is the Authenticated Users group. When a computer joins a domain, the Domain Users group is added to the Users group on the computer.

S-1-5-32-546: Guests

> A built-in group. By default, the only member is the Guest account. The Guests group allows occasional or one-time users to log on with limited privileges to a computer's built-in Guest account.

S-1-5-32-547: Power Users

> A built-in group. By default, the group has no members. Power users can create local users and groups; modify and delete accounts that they have created; and remove users from the Power Users, Users, and Guests groups. Power users also can install programs; create, manage, and delete local printers; and create and delete file shares.

S-1-5-32-548: Account Operators

> A built-in group that exists only on domain controllers. By default, the group has no members. By default, Account Operators have permission to create, modify, and delete accounts for users, groups, and computers in all containers and organizational units of Active Directory except the Builtin container and the Domain Controllers OU. Account Operators do not have permission to modify the Administrators and Domain Admins groups, nor do they have permission to modify the accounts for members of those groups.

S-1-5-32-549: Server Operators

> A built-in group that exists only on domain controllers. By default, the group has no members. Server Operators can log on to a server interactively; create and delete network shares; start and stop services; back up and restore files; format the hard disk of the computer; and shut down the computer.

S-1-5-32-550: Print Operators

> A built-in group that exists only on domain controllers. By default, the only member is the Domain Users group. Print Operators can manage printers and document queues.

S-1-5-32-551: Backup Operators

> A built-in group. By default, the group has no members. Backup Operators can back up and restore all files on a computer, regardless of the permissions that protect those files. Backup Operators also can log on to the computer and shut it down.

S-1-5-32-552: Replicators

> A built-in group that is used by the File Replication service on domain controllers. By default, the group has no members. Do not add users to this group.

Additional Resources

List of well-known SIDs: http://support.microsoft.com/kb/243330

SID & GUID converter utility: http://www.sekchek.com/SekCheklocalsw.htm (SekChek software)

- The utility displays an object's SID in binary, hexadecimal and SDDL formats.

SID resolver utility: http://www.sekchek.com/SekCheklocalsw.htm (SekChek software)

- The utility resolves a SID and displays the object's friendly name, e.g. MyDomain\MyAccount.