

Re: windows services question

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2006-12/msg00487.htm>

- *From:* "anton bassov" <soviet_bloke@xxxxxxxxxxx>
 - *Date:* 19 Dec 2006 07:31:18 -0800
-

I therefore don't follow your point about why it's enabled by default in the system account but not for ordinary administrators.

This is how everything sets configured upon the setup, so this is a default. Certainly, Admins can change everything at some later point, and add and/or enable any privilege you wish, so that even restricted users may have SeDebug privilege in the token. However, I am just speaking about defaults..

there's nothing special about the "LocalSystem" account in this regard.

... apart from the fact that it is able to access SAM database, "\\Device\\PhysicalMemory" on OS versions below W2K3 SP1, etc, etc, etc by default, but Admins have no chance to do so. Again, Admins can grant themselves access to these items (they can do it programmatically right on the fly), but this is already not a default setting, is it???

If an Administrator turns it on however then the Task Manager will respect it and kill whatever you want.

Sure. However, again, this is not a default

Basically, I am speaking solely about default configurations – it is understandable that Admins are able to change everything the way they like

Anton Bassov

Re: windows services question

Larry Smith wrote:

This is wrong....

Only LocalSystem and Admins have this privilege in their tokens, in the first place. It is disabled for Admins by default, but enabled for LocalSystem, because the system must be able to open any process for any access....

No disputing that nor did I say otherwise. You can add this privilege to anyone's account if you wish however though it's not something that anyone would normally do obviously. Once a privilege does exist in your token however, it's easily enabled on-the-fly and in fact, some WinAPI functions quietly do this behind the scenes (enabling and then disabling it again if the privilege is required for that function). IOW, there's nothing special about the "LocalSystem" account in this regard. A program running as a member of the Administrators group can easily enable SeDebugPrivilege if it so chooses. The Task Manager chooses not to in order to protect people from themselves including Administrators. I therefore don't follow your point about why it's enabled by default in the system account but not for ordinary administrators. It's not simply because the system account needs to open any process for all access which of course it must do. That is, an ordinary administrator can do this as well if it wants (by simply enabling the privilege on-the-fly). The reason it's enabled by default for the system account is presumably for efficiency. It needs this privilege regularly so MSFT turned it on by default. That seems the most reasonable explanation anyway (IMO) noting that not all privileges are on by default even for the system account (which seems to back up my point).

system account is also an administrator BTW

This is wrong as well....

No it's not. The administrator's group is in the system account's token and always has been. Check this out for yourself under the system account's SID (S-1-5-18). You'll find group SID S-1-5-32-544 which is BUILTIN\Administrators.

The trick is to enable it first

Exactly, but Task Manager does not do it – this is why you cannot terminate a process that runs under the LocalSystem account, via it...

Re: windows services question

As I basically said. If an Administrator turns it on however then the Task Manager will respect it and kill whatever you want.

However, if some third-party app does it, there is nothing that OP can do about it

Agreed, since you can't stop an administrator from doing anything they really want without starting an (unwinnable) administrative war.