

Re: How to work around UAC?

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.platformsdk.security/2007-10/msg00026.html>

- *From:* "Chris Becke" <chris.becke@xxxxxxxxx>
 - *Date:* Wed, 17 Oct 2007 09:36:31 +0200
-

for all users of the system, including some system processes. The problem is that in some situations our DLLs are also loaded for non-privileged user applications. In these situations, and with UAC active, our DLLs no longer have access to our software's registry keys. Since having separate copies of our registry data for each user that may log onto the system is not feasible, where should our software store its registry data?

Ok there are some mitigations here.

First off, your DLLs DO have access to HKLM – but read only. Your basic problem is you are using a all_access type flag when opening the keys that you are ultimately going to read from.

And you are just going to read settings in those keys right?

normal "non-privileged" users normally have no place setting systemwide settings.

If – and only if – you DO need normal users to write to HKLM, AND the result of doing so will not screw over other users in any way, then you can use the security APIs to change the access control list on subkeys within HKLM. You can. It is allowed for applications to change the access restrictions on their own sub-portions of HKLM.

However, a better solution to this problem would be to construct an API by which your DLL could communicate with a service component. Your service would do the actual writing to HKLM after validating that the non-privileged users request to write is sane.

That would solve many of our problems, but since our registry keys are stored under HKEY_LOCAL_MACHINE, is that possible? If our installer did modify the ACLs for the stuff we need, how would that effect UAC?

The registry is just like the file system in terms of how it is protected with ACLs. You can use the 'regedt32.exe' too to view the permissions for

Re: How to work around UAC?

each key and will see that HKCU will have write access (full control) for the user, while HKLM, by default, gives the Administrator full control, but 'everyone' gets Read only.

When your setup program – running as administrator – creates keys under HKLM it can set the ACLs at that time for the created keys to give 'everyone' full control.

"Mick" <Mick@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:88AF3BA2-80D4-424A-AF8A-F5CFFE873E7C@xxxxxxxxxxxxxxxxxxxx

"Chris Becke" wrote:

Search for WindowsVistaUACDevReqs.doc

You should get a download page following this link:

<http://www.microsoft.com/downloads/details.aspx?familyid=ba73b169-a648-49af-bc5e-a2eebb74c1>

I downloaded the document. Now, if I could only understand it.

To summarize:

You are going to need to manifest all *your* apps as this will prevent virtualization, and it will properly cause "Administrator" prompts to appear for any apps that need to write to system locations.

Legacy apps that don't use manifests will be able to read data in the common system locations, but might virtualize if they attempt to write.

To bypass this, either ensure that the legacy apps get updated with manifests (you *CAN* just drop a appname.exe.manifest into directory containing a legacy app to disable virtualization in that app.)

Okay, so our applications can use manifests to turn off virtualization, but that won't work for DLLs, right?

One of our problems is that our DLLs are designed to store configuration data under HKEY_LOCAL_MACHINE\SOFTWARE\OurSoftware. Our understanding was that this was how it was supposed to be done because our software is intended for all users of the system, including some system processes. The problem is that in some situations our DLLs are also loaded for non-privileged user applications. In these situations, and with UAC active, our DLLs no longer have access to our software's registry keys. Since having separate copies

Re: How to work around UAC?

of
our registry data for each user that may log onto the system is not
feasible,
where should our software store its registry data?

Once an app has virtualization disabled, it will get access denied errors trying to access protected resources in inappropriate ways (Rather than being virtualized) so you need to consider getting your installer to modify the ACL for the registry keys and folders containing stuff that needs to be writable by non administrator mode components.

That would solve many of our problems, but since our registry keys are stored under HKEY_LOCAL_MACHINE, is that possible? If our installer did modify the ACLs for the stuff we need, how would that effect UAC?

Thanks for your help Chris! It is much appreciated.