

Re: Deny _WRITE_ access to a file

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.security/2005-03/0405.html>

From: Javier J (*no.mail_at_please.no*)

Date: 03/10/05

Date: Thu, 10 Mar 2005 21:53:24 +0100

On Wed, 2 Mar 2005 11:39:41 -0700, "Al Dunbar [MS-MVP]"
<alan-no-drub-spam@hotmail.com> wrote:

>
> "Javier J" <no.mail@please.no> wrote in message
> news:vpf921h1j3tgiliegad4u4kj7e7urhl72l@4ax.com...
>> Hi!!
>>
>> Thanks a lot for the response.
>
> You are welcome...
>
>> First of all, regarding LOGON SCRIPT, the mistake is mine: What I was
>> trying to talk about was a STARTUP script (if I'm not mistaken, that
>> script runs as BUILTIN\SYSTEM).
>>
>> I think I'd rather explain a bit more about the environment so that
>> it's clear of why I'm asking for such strange things:
>>
>> The situation is as follows: The PCs in question (Win 2000 PRO, SP4+,
>> W2000 Mixed Domain) "belong" to a group of users who, as part of their
>> normal duties, have to handle sensitive information using an internal
>> company app. To avoid undue information leakage, these users have
>> *TWO* logon users for the domain, a highly restricted one that is used
>> to run the corporate app/access sensitive information, and a "Normal"
>> user for the rest of everyday tasks.
>
> Interesting. In our environment, any exceptionally privileged accounts are
> restricted in what they can do only by policies written on paper rather than
> being enforced by GPO. Admin accounts are not allowed to use the browser,
> but they are not actually prevented from doing so. We do not have any
> applications that need to be run in a context where the rest of the
> infrastructure is made inaccessible, so access to such apps is granted to
> normal user accounts.
>

Hi!

In the "normal" running of this org, that is the way things are. But this is a "small" group (around 50 users, max) that sometimes handle extremely sensitive (from the legal P.O.W.) info, so we have decided to try to go the "extra mile" and "really" restrict the things.

Of course, the legal "threat" is still in place, but the consensus was that the user should also be "made aware" of the restricted nature of the environment... If there is plenty of things that the user normally can do that can't be done, the user can't (honestly, justifiably) say "I was not aware of that" ...

>> *The "normal" user can run all software EXCEPT the restricted app, and can work normally.*

>>

>> *The setup for the restricted user (using GPO, crypto software et al) is such that the restricted user only can run the "sensitive" app, they can't browse or "see" in Explorer the local folders, their profile is redirected to an encrypted network etc etc...*

>

>... *still having some trouble envisioning how this type of operation can be configured without creating complications for administration. I mean, who here can list all of the files on a Windows system to which ANY user MUST have READ access? WRITE access?*

Well, basically it's not that difficult.

Normal users only need read access to %windir% and to %programfiles% (and that's the default windows user perms), and write access to their own user profile.

So any other folder (for example, those hanging from C: that are _not_ c:\winnt or c:\Program Files or c:\Documents and Settings) are the folders I worry about.

As for the temporary folders, and swap file, and such, the system is set to clear them on shutdown, and the user profile can't "close session", only "restart".

>> *Also, using an STARTUP batch script,*

>

> *Fine. But what if the system is not restarted between user sessions?*

>

Well, those policies are applied _each_ time the system boots, but a normal user shouldn't be able to change them (as they're not members of the admin group), so it's probably a bit of overkill to have the policies re-set on each machine boot...

On user logon and logoff there is a more limited script that performs temporary file deletion and such niceties as we could think of...

Also, we are exploring the possibility of forcing the user to reboot the computer when they logoff the "restricted user", but that's only on the drawing board at the moment.

>> *the members of the restricted group have been DENIED access to different .exes that restricted users should not run (ftp.exe, telnet.exe and other) and folders they don't need access to. (Windows already protects system folders against accidental change). The problem is, there are a couple of folders on C:\ (such as c:\local_settings) that the user logon needs to be able to read, because it sets machine-specific config. (such as the building's mail server, the NT server, and suchlike)*

>>

>> *The problem is that the folder is set to be writeable by "Everyone". I'd like to be able to "change" it so "no write" for the users of this particular group. I can DENY access, but these users are part of "Everyone", so even if "RestrictedG" has only READ access, as they are members of "Everyone"; they get to write there...*

>

>*The explicit DENY will generally override the ALLOW.*

That's why I'm trying to "get" how to be able to selectively DENY some things, as I'd rather not deny everything :)

>

>> *Why am I exploring the "deny" route, instead of limiting the rights of "Everyone".. because there are some cases where the normal users has to be able to write, so "Everyone:W" is a valid permission.... as long as I could do something like "RestrictedG":DENY WRITE....*

>>

>> *I know that permission is "settable" (is that a word?) as it can be set using (the "simple") NTFS Perms. tab... but to script it is what is driving me crazy!!*

>

>*I sympathize! But I fear that the approach may not be as bullet proof as you appear to need it to be.*

Well.... I am aware that there is no perfect solution to the problem. What I'm trying to do is to configure the system so it's *notably* (at least, from my POV) more secure than it was, and to do so in such a way that the user's productivity does not suffer too much...

>

>*Apparently members of the restricted group can logon and establish their redirected session having only READ access to the "c:\local_settings\" folder. If it is possible to script a permissions change such that the folder remains read/write for everyone except for the restricted group who have read-only access, then surely it should be possible to apply that permission setting ONCE using the GUI, and just leave it at the setting you need.*

>

>I tried this and it seemed to work, however, I had to put a checkmark beside
>WRITE in the DENY column. When I tried to DENY the MODIFY access, I found I
>could not do so without also denying READ, READ&EXECUTE, and LIST FOLDER
>CONTENTS. That would appear to be a limitation or constraint applied by the
>GUI itself.
>
>I would recommend that you try the following from the GUI: Everyone:W and
>RestrictedG:DENY WRITE, and then test the results to see if it achieves your
>purpose without causing problems for the restricted users.

YES! That's the final result I want to achieve, but using a script. I
already have tested those changes on the GUI, the "problem" is getting
the computers to that state w/o having to log-on to them.

>If that fails in any way, then I would suggest drilling down to the advanced
>tab and denying everything associated with being able to modify or write.
>
>If that fails in any way, then I would suggest playing with cacls or
>cacls.vbs to see if the extra granularity allows you to do what you want.

No, I only need to be able to do the same than in the "standard" file
permission tab, BUT w/o doing it manually.

>If THAT fails, then, IMHO, you will be unable to accomplish this in a
>startup script. If it succeeds, then you will not need to do it in a startup
>script if you can configure the permissions of that folder as a default in
>your image, or as part of the restricted application installation process.
>

The issue here is that the install image is already "closed", and it
depends on another dept. that I don't have "leverage" with. And that
installation image could change with time (for instance, when new
hardware is bought). What I am trying to do is to find a way to
"transform" the image from the permissions-state that it has, to the
one I want it to have, so that I can be confident that "accidents"
don't happen (say, one of the Domain Admins logs on the computer, and
makes some changes, or whatever). That's why I'm investigating
software installation policies, permission change scripts and the
like..

The idea is to develop a "method" to be able to securize a given
computer, not only a "lock this specific workstation" approach. And to
minimize the manual intervention required. (as it is, it's still
necessary to log on to the computer as local admin at least once, so
that MS OUTLOOK is completely installed.... And I don't know why
should it behave like that, it's.. odd and uncomfortable, to say the
least)... BTW, if you know of any way of avoiding that, I'd be more
than happy to hear it!! ;)

>/Al
>

microsoft.public.security: Re: Deny _WRITE_ access to a file

>> *Thanks a lot. Any help _WILL_ Be more than welcome!!*

>>

>> *Javier J*

>>

>> *On Mon, 28 Feb 2005 22:12:30 -0700, "Roger Abell" <mvpNOspam@asu.edu>*

>> *wrote:*

>>

>> *>Al is quite right in picking up on your mention of use in a*

>> *>login script – which skipped my attention.*

>> *>To do as you had planned you would need to do this in*

>> *>a startup/shutdown script, not login/logoff script.*

>> >

>> *>However, you really, really would IMO be better off by*

>> *>restructuring so that all files with this requirement are in*

>> *>a folder with appropriate grants, not mixed in with other*

>> *>files in a folder where the default NTFS permissions will*

>> *>need to be changed.*

>>

>