

Re: Unable to prevent OU deletion by Domain Admins?

Source:

http://www.tech-archive.net/Archive/Win2000/microsoft.public.win2000.active_directory/2004-11/0325.html

From: Mark Renoden [MSFT] (*markreno_at_online.microsoft.com*)

Date: 11/04/04

Date: Fri, 5 Nov 2004 08:41:28 +1100

Hi

As much as what Paul is saying should assist you in achieving your goal, you should avoid doing this. It's easy to break things when you start changing the default rights and permissions of administrators and if anyone else ends up owning the administration of this and wonders why something won't work as expected, it'll take considerable effort to understand why.

Create a new group and enable it to do only the things you want it to. Put people who can't be trusted into this group and all is well. You might only leave yourself and the Administrator account in the administrators group. Much much safer and much much simpler to manage.

Kind regards

--

Mark Renoden [MSFT]

Windows Platform Support Team

Email: markreno@online.microsoft.com

Please note you'll need to strip ".online" from my email address to email me; I'll post a response back to the group.

This posting is provided "AS IS" with no warranties, and confers no rights.

"ptwilliams" <ptw2001@hotmail.com> wrote in message

news:e3o86cewEHA.3376@TK2MSFTNGP12.phx.gbl...

>A couple of points if I may...

>

>> Not sure what you mean by your "other post," but what I'm pointing out is

>> that the behaviour is improper.

>

> I made another post earlier on; perhaps you cannot see this -I've been

> seeing this behaviour when posting through the communities website,

> sometimes the posts don't show up when I access these groups from a

> newsreader...

>

>

>> It is even worse when Microsoft's own guidelines for parsing ACLs states

>> that DENY ACLs trump any allow ACLs

>

> That's not entirely accurate. Deny permissions take precedence over allow

> permissions with one exception: inherited deny permissions don't take

microsoft.public.win2000.active_directory: Re: Unable to prevent OU deletion by Domain Admins?

```
> precedence over non-inherited allow permissions.
>
> The SRM (Security Reference Monitor) evaluates an access request by
> parsing
> the list of permission entries in the DACL. It does this starting at the
> top of the list and working its way down until it finds enough entries to
> allow or deny permission. Which means that there is an order to what the
> effective permission are. This is as follows:
>
> Non-inherited deny entries, non-inherited allow entries, inherited deny
> entries and inherited allow entries.
>
>
>> If you explicitly remove permission inheritance on an object, then
>> explicitly set DENY ACLs on that object for a group, then the denied user
>> or group should not be able to perform the action that the ACL forbids.
>> That's how ACLs work, or at least how they are supposed to work.
>
> The problem here is that the Delete Subtree permission isn't inherited
> per-se. That is, this is a folder (container) permission not a file
> (child)
> permission. Which means even if inheritance is not enabled, this
> permission
> on the parent container is still in effect. If you consider what I've
> said
> above, the deny permission set lower down doesn't come into it because the
> right at a higher level is enforced.
>
> However, bear in mind that the delete subtree permission goes hand-in-hand
> with the delete permission on the root level of the subtree.
>
> D
> |
> -- OU
>   |
>   -- obj
>   |
>   -- obj
>
> You could modify the default domain admins permissions so that they no
> longer have the Delete subtree permission. Or you could remove the delete
> permission from the domain level, and reintroduce it lower down.
>
> Delete Subfolders, and delete at the D level will delete all underneath.
> I
> believe this was implemented for POSIX compatibility.
>
>
> (Hopefully I've got all that right ;-)
```

```
> --
>
> Paul Williams
>
> http://www.msresource.net
> http://forums.msresource.net
>
> "Josh" <joshuabrown@gmail.com> wrote in message
> news:e7d88c81.0411030934.7977ae61@posting.google.com...
> Paul,
```

microsoft.public.win2000.active_directory: Re: Unable to prevent OU deletion by Domain Admins?

> Not sure what you mean by your "other post," but what I'm pointing out
> is that the behavior is improper. If you explicitly remove permission
> inheritance on an object, then explicitly set DENY ACLs on that object
> for a group, then the denied user or group should not be able to
> perform the action that the ACL forbids. That's how ACLs work, or at
> least how they are supposed to work. It is even worse when
> Microsoft's own guidelines for parsing ACLs states that DENY ACLs
> trump any allow ACLs; when you look at the ACLs on an object, it even
> appears that you have accomplished what you want to
> accomplish--protecting the object from deletion.
>
> I understand that domain admins have the delete and delete subtree
> rights at the domain level. But think of this in terms of files and
> folders: I have a folder where Domain Users have Full control rights.
> Inside that folder I have a file, and onto that file I place a Deny
> Delete ACL for Domain Users (or even for just a single user who is a
> member of the Domain Users group). That ACL works as advertised, and
> will deny that user/users the ability to delete the file. There
> shouldn't be anything about a group--even a builtin group--that
> overrides local security settings like that.
>
> This isn't about trusting an admin. It's about the possibility that
> something could be accidentally deleted (everyone makes mistakes,
> especially during after hours recovery sessions, etc.). We should be
> able to protect against that possibility, and Microsoft is removing
> control from our hands and ignoring explicit security settings in a
> very odd way.
>
>
> "ptwilliams" <ptw2001@hotmail.com> wrote in message
> news:<O6ioUWRwEHA.1396@tk2msftngp13.phx.gbl>...
>> As I said in my other post, this behaviour occurs because the domain
>> admins
>> group has the delete and delete subtree permission at the domain
>> level -which overrides lower-level container object permissions -just
>> like
>> NTFS file and folder permissions.
>>
>> Trust is the issue here. And if anybody make the mistake of deleting an
>> OU
>> twice...well....
>>
>> Where's the change control processes, eh?? <g>
>>
>> --
>>
>> Paul Williams
>>
>> <http://www.msresource.net>
>> <http://forums.msresource.net>
>>
>> "Josh" <joshuabrown@gmail.com> wrote in message
>> news:e7d88c81.0411020810.12f124a3@posting.google.com...
>> Mark,
>>
>> Thanks, but that doesn't really answer my question. We have a
>> situation where we want to prevent a particular OU from being
>> accidentally deleted. Trusting our domain admins doesn't prevent them
>> from making very human mistakes. This looks like a bug to me--why can
>> I not create an OU that denies deletion rights to domain admins, when
>> the ACL appears that it should do precisely that?
>>

microsoft.public.win2000.active_directory: Re: Unable to prevent OU deletion by Domain Admins?

>> I challenge anyone to try what I have outlined above and get it to
>> properly deny deletion rights. If this right is not working properly,
>> how am I to have confidence in any of our settings? Deny rights is as
>> important as allow rights, if not more so, since deny rights are
>> supposed to trump allow rights.
>>
>> Josh
>>
>> "Mark Renoden [MSFT]" <markreno@online.microsoft.com> wrote in message
>> news:<uP56r\$FwEHA.1988@TK2MSFTNGP12.phx.gbl>...
>> > Hi Josh
>> >
>> > It's better practice to give rights to a group of users rather than
>> > take
>> > them away from Domain Admins. You should never alter the rights of a
>> > builtin group or user. You're better off creating a group for the
>> > purpose
>> > of administering OU's, delegating permissions to this group and keeping
>> > your
>> > Domain Admins group to a very select few that can be trusted.
>> >
>> > Kind regards
>> > --
>> > Mark Renoden [MSFT]
>> > Windows Platform Support Team
>> > Email: markreno@online.microsoft.com
>> >
>> > Please note you'll need to strip ".online" from my email address to
>> > email
>> > me; I'll post a response back to the group.
>> >
>> > This posting is provided "AS IS" with no warranties, and confers no
>> > rights.
>> >
>> > "Josh" <joshuabrown@gmail.com> wrote in message
>> > news:e7d88c81.0411011409.cld654b@posting.google.com...
>> > >I am trying (unsuccessfully) to prevent accidental deletion of several
>> > > OUs by our domain admins. For testing purposes, I have done this:
>> > >
>> > > 1) Create new OU, removed inheritance of permissions.
>> > > 2) Removed all groups from the permissions
>> > > 3) Added Domain Admins with Full Control
>> > > 4) Explicitly set Deny rights for Domain Admins for Delete, Delete
>> > > Subtree, and Delete Organizational Object.
>> > >
>> > > Create new user, add user to Domain Admins. Log in with user, and
>> > > the
>> > > OU can be deleted without warning.
>> > >
>> > > The only way I have gotten this to work is by creating a user in the
>> > > OU that I want to protect, and setting Deny All rights for the Domain
>> > > Admins group on that user. That prevents Domain Admins from deleting
>> > > the parent OU, but it is a pretty bad solution...and it doesn't
>> > > explain why the Domain Admins can delete the OU when all relevant
>> > > deletion ACLs are set to Deny.
>> > >
>> > > Any thoughts?
>
>