



Published on Symantec Juice (<http://www.symantec.com/community>)

Group Policies: Applying to Specific Users

By *trb48*

Created 15 May 2007 - 11:35



[1]

In my [last article](#) [2], I talked about the advantages of using Group Policies in an image. What if you want the policies to apply to all users except the Administrators? This is the guide for you.

Setting Everything Up

Preparation

The first thing that we need to do is set aside a test computer where we can set up the Group Policies just the way we want them. Why? I cannot tell you how many times I have been setting up Group Policies (while I was creating an image) and I have locked down the machine so much that I have not been able to do anything (as an Administrator that is).

On the test computer make sure you are using the same image that you are using out in the field. Also, make sure that the account that you set the Group Policies in is the same as the average users account in your environment.

One last thing to keep in mind, make sure that you have access to the CACLS. We will use this command line program later in the article.

Setting the Group Policies

We are now ready to set some Group Policies. To open the Group Policy Editor, do the following:

Go to Start >> Run, one the Run window opens, type in "GPEDIT.MSC" without the quotes.

The Group Policy window will open. I spend most of my time in User Configuration >> Administrative Template.

Setting Group Policies will take some time. Why? There are tons of settings. Look at each one and if you find a policy that you think that you want to enable (or disable) double click on it. Then go to the "Explain" tab to get more information.

Note: Make sure that you spend some time in User Configuration >> Administrative Templates >> Windows Components >> Microsoft Management Console >> Group Policy. You will probably want to enable "Group Policy Management." Why? After you enable this policy (and close the Group Policy Editor) you can no longer access the Group Policy Editor. That is a good thing. If you are setting Group Policies you probably don't want the end user to turn around and disable them.

At this point the Group Policies are applying to all accounts. If you are okay with every account having the same Group Policies. If you don't want the Administrator account to be locked down, keep on reading.

Excluding the Administrator Account

Group Policies are great, but they can get annoying in the Administrator's account. If I can't even use the Control Panel (which is disabled for a normal user in my standard image), I get really annoyed. There is a simple way to prevent the Administrator account from getting the Group Policies. Running this simple script will solve the problem:

The Code

```
REM Set File Permissions
echo y| CACLS "C:\WINDOWS\system32\GroupPolicy\Machine\Registry.pol" /D Administrator
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\Machine\Registry.pol" /e /g Administrators:f
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\Machine\Registry.pol" /e /g System:f
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\Machine\Registry.pol" /e /g "Authenticated Users"

echo y| CACLS "C:\WINDOWS\system32\GroupPolicy\User\Registry.pol" /D Administrator
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\User\Registry.pol" /e /g Administrators:f
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\User\Registry.pol" /e /g System:f
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\User\Registry.pol" /e /g "Authenticated Users"

echo y| CACLS "C:\WINDOWS\system32\GroupPolicy\gpt.ini" /D Administrator
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\gpt.ini" /e /g Administrators:f
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\gpt.ini" /e /g System:f
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\gpt.ini" /e /g "Authenticated Users":r

REM Set Folder Permissions
echo y| CACLS "C:\WINDOWS\system32\GroupPolicy" /D Administrator
echo y| cacls "C:\WINDOWS\system32\GroupPolicy" /e /g Administrators:f
echo y| cacls "C:\WINDOWS\system32\GroupPolicy" /e /g System:f
echo y| cacls "C:\WINDOWS\system32\GroupPolicy" /e /g "Authenticated Users":r
```

The great thing about this script is that it can be run using a "Run Script" deployment task (in Deployment Console). So, if you already have the group policies deployed, you can send this right on out. If you are building an image, just run this script after you are done setting the Group Policies. What is this script doing? Here is an explanation:

Denying the Administrator Access

In the script above, I go through each file found in the Group Policy folder and change the security settings. How? I change which users have access to the account. As you can see, each file has four lines of code associated with it. Lets talk about the tags that I used:

echo y|

This part of the code is only used in automated scripting. Anytime you run a CACLS command, it asks you if you want to proceed. Including the "echo y|" at the beginning of the scripting, it answers the "Are you sure (Y/N)?" question in the affirmative.

CACLS

CACLS is a powerful command-line tool that is used for setting user options. For more information on the CALCS command, Microsoft has prepared a very useful guide. You can find it here: [CALCS](#) [3]

Another useful way to find out what options you can set with any command-line go to Start >> Run,

and type in "CMD" (with out the quotes). The command prompt will open. Now type in "CALCS /?" (again, without the quotes). It will look like this:

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 5.0.2600.5512]
(c) Copyright 1985-2004 Microsoft Corp.

C:\Windows\system32>CALCS /?

Usage of wildcard access control files (ACLS of files)

Syntax: CALCS [/D] [/G] [/E] [/F] [/R] [/W] [/P] [/S] [/I] [/M] [/O]
           [/C] [/V] [/Q] [/L] [/N] [/D] [/G] [/E] [/F] [/R] [/W]
           [/P] [/S] [/I] [/M] [/O] [/C] [/V] [/Q] [/L] [/N]
           [/D] [/G] [/E] [/F] [/R] [/W] [/P] [/S] [/I]
           [/M] [/O] [/C] [/V] [/Q] [/L] [/N]

Options:
  /D Deny specified user's access rights (only valid with /O).
  /G Grant specified user's access rights (only valid with /O).
  /E Edit specified user's access rights.
  /F Specify full control.
  /R Specify read access.
  /W Specify write access.
  /P Specify permissions.
  /S Specify permissions for subdirectories and files.
  /I Specify inheritance.
  /M Specify mode.
  /O Specify owner.
  /C Specify cache.
  /V Specify verbose.
  /Q Specify quiet.
  /L Specify local.
  /N Specify network.

Remarks:
  The ACLS files will be inherited by subdirectories.
  The ACLS files will be inherited by files.
  The ACLS files will apply to the current file/directory.
  The ACLS files will apply to the current file/directory.
  
```

[4]

[Click to view.](#) [4]

File Path

At the next part of the script, we find the file path. This tells the CALCS command where the file is that we want to change the settings on.

/D UserName

If you use this tag, you deny the specified user access to the file. In the scripting above, you can see that I have denied the Administrator access to the file.

Note: You can use a group name instead of a user name. So, I could deny the "Administrators" group if I wanted to.

Bringing it all Together

```
echo y| CACLS "C:\WINDOWS\system32\GroupPolicy\Machine\Registry.pol" /D Administrator
```

Granting Access to other users

Most of the commands are the same, so, lets talk about some new commands that grants access to the file:

/e

This tells the CALCS command to edit the users access instead of replacing it. Since the user already has access to the file, editing will do for us.

/g UserName:permission

The /g command grants the specified user access to the file. We also have to tell the CALCS command what type of access we want the user to have. In the example above, used *f*. That gives the user full access. You can also use:

- **r** - Read Access
- **w** - Write Access
- **c** - Change (write)
- **f** - Full control

Note: I am not using a "UserName" in this example. I am using a group name. You can use them interchangeably.

Bringing it all Together

Now we know how to give specific access to our users. In the example below, I give the Administrators group full access to this file.

```
echo y| cacls "C:\WINDOWS\system32\GroupPolicy\Machine\Registry.pol" /e /g Administrators:f
```

Now, if we run the code above (look above at "REM Set File Permissions") in a bat file or through Deployment Solution every account will get the Group Policies that we have set, except the Administrator account.

I really like this solution. As a system admin I want access to every Windows tool so I can do my job. But, I don't want the end user to have access to every window tool. This solution allows us the best of both worlds.

Deploying the Solution

What if you have an image that has already been deployed, and now you want to use Group Policies. One solution is to create a new image and deploy it. I know what you are thinking, and I agree. That would be way to much work. In this last section, I will show you how to deploy a set of Group Policies.

Building a Package

To make things simple, I am going to make a RIP. If you have purchased Deployment Solution, you have access to RInstall.

- Open RInstall on the system that you have set the Group Policies on.
- Now, in the left pane (on the bottom) under Drives right click
- Go to New >> Existing Item
- Now, navigate to C:\Windows\System32
- Select the GroupPolicy folder and click the "OK" button.

Now the files are inside the RIP.

Scripting

In order to import these settings into a new machine we have to open the Group Policy Editor before we copy these files. To do this, create a bat file with the following script inside:

```
"C:\Windows\System32\gpedit.msc"
```

Now do the following:

- Go to Edit >> RIP Options
- Check the box next to "Run without user interaction"
- Click on the "Details" button
- Under the "Pre Install Scripts" click on the "Add" button
- Click on the "Browse" button, and find the bat file we created earlier.
- Now, click the box next to "Add File to RIP"
- There is a drop down menu in the middle of the "Add Script" window, select "Hidden"
- Click the "OK" button

Now, I would like to set the file permissions of these files so that the Administrator account is Group Policy free. I will create a second bat file that has the scripting found above (found in "The Code" section). Now we should add that bat file to the RIP. To do that, do the following:

- Go to the "Post Install Script"
- Click on the "Add" button
- Click on the "Browse" button, and find the bat file we just made.
- Now, click the box next to "Add File to RIP"
- There is a drop down menu in the middle of the "Add Script" window, select "Hidden"
- Click the "OK" button

Lets Finish Making the RIP

- In the Options window, click the "OK" button
- In the original Options window, click on the "OK" button when you are all done
- Save the RIP, and you are done!

So, when this file runs it will do the following:

- Open the Group Policy Editor (which makes it possible to import Group Policies from other computers)
- Copy the Group Policy files that we set up on our test computer
- Set the file permissions on the Group Policy files to exclude the Administrator

Once this RIP is created, you can deploy this package through Deployment Solution. Now you can push out Group Policies to your computer, and almost instantly your systems will be more secure.

[Automation](#) [Best Practices](#) [Configuration](#) [Deployment](#)
[Newsletter](#) | [Submit Content](#) | [Get RSS](#) | [Contact Us](#) | [© 2008 Symantec, Inc. All rights reserved.](#)

Source URL: <http://www.symantec.com/community/node/1624>

Links:

- [1] <http://www.symantec.com/community/node/1624>
- [2] <http://www.symantec.com/community/node/1623>
- [3] <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/cacls.mspx?mfr=true>
- [4] http://www.symantec.com/community/sites/default/files/img/1624_0.jpg