Discovering What Your Users Have Access To – Permissions

Abstract

If you've ever faced a "simple" task like determining exactly what resources a given employee has access to across all your servers, you know that permissions management can be tedious, time-consuming, and error-prone. Answer-ing just one question like that can easily take a week of work with native tools. But there is a better way. This white paper explains the permissions management problem – and its solution.

Introduction

"Brian was just fired. We disabled his account. Can you tell me what he had permission to on all of the servers?"

No administrator likes to hear that question. You have two possible responses:

- Laugh outrageously
- Weep uncontrollably

You're certainly not going to be able to answer the question. Here's another classic:

"Hey, those master project files? The ones we have living on one file server in each of our eight offices? We need to change the permissions on those so that the Interns user group doesn't have access anymore. When can you be finished with that?"

- Never.
- It's already done! I used sparkly fairy magic!

Face it, Windows permissions were easy to manage when you had one or two servers. Not so much with a dozen or a hundred. Microsoft knows it, too, and is furiously re-thinking the entire way that permissions are managed to try and address the problem. Of course, until they figure it out and get a fix in place, we're all stuck with the old way.

Let's answer the permissions questions!

Let's see what it would actually take to answer these two questions. Let's assume – just to keep things simple – that we're only talking about files and folders. That won't always be the case, of course; normally, you'd also have to consider registries, printers, services, scheduled tasks, SharePoint, SQL Server, Exchange, and more. But let's just focus on files for a second..

If you've ever faced a "simple" task like determining exactly what resources a given employee has access to across all your servers, you know that permissions management can be tedious, time-consuming, and error-prone.

Discovering what Brian had access to

To answer the first permissions question (what Brian had access to), you're going to need Brian's security identifier, or SID. Windows PowerShell can do that for you:

```
Import-Module ActiveDirectory
Get-ADUser Brian
DistinguishedName : CN=Brian,OU=Sales,DC=company,DC=pri
Enabled : False
GivenName :
Name : Brian
ObjectClass : user
ObjectGUID : 7cbb90e3-594e-4e53-837a-a4b4d40896f4
SamAccountName : Brian
SID : S-1-5-21-29812541-3325070801-1520984716-1108
Surname :
UserPrincipalName :
```

Okay, there we are: S-1-5-21-29812541-3325070801-1520984716-1108. Now you're going to need a copy of the icacls.exe utility. On your first file server, go to the root of the C: drive and run this:

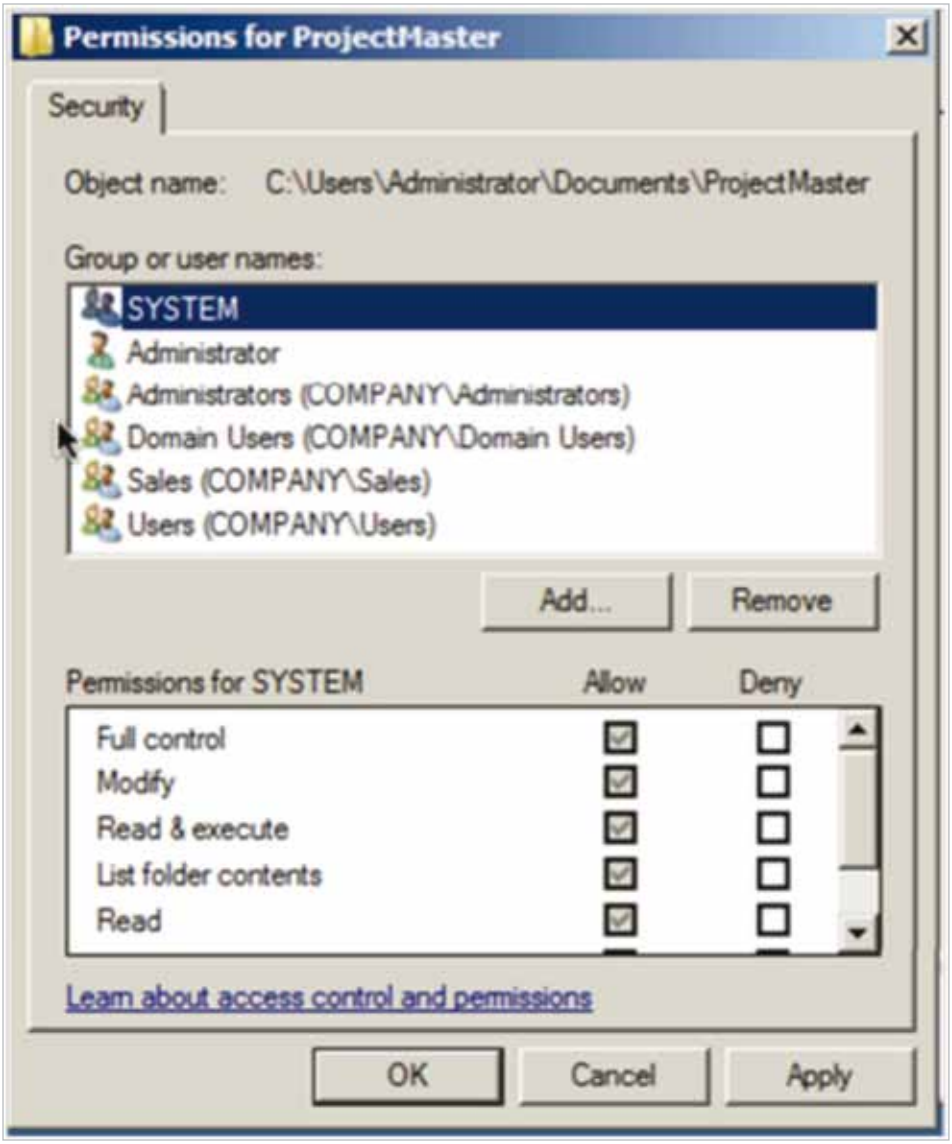
```
Icacls * /findsid S-1-5-21-29812541-3325070801-1520984716-1108 /T /C
```

Now sit back and wait. Actually, don't wait; instead, open up a new console window and start running it on every other drive on the server. Then, "lather, rinse, repeat": head to every other server you need to check and start running the command on each of them as well. It's going to take a long time: Windows has to check each and every file, one at a time, to see if the SID indicated appears there.

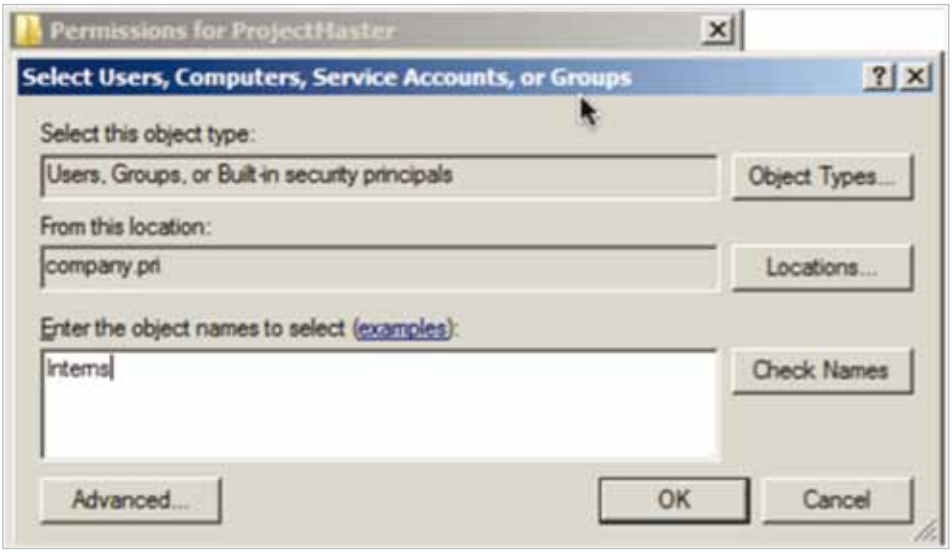
And that will only tell you the files and folders where Brian's been granted direct permission – it won't tell you what he's had access to via group membership. To do that, you're going to have to start over, running icacls again and again and again, each time using the SID from one of the groups Brian belonged to. You should be done in a week or so.

Removing access for the interns group to the master project files

What about the re-permissioning request? Let's tackle that the GUI way, since you might be feeling a little disappointed in icacls (although it could totally handle this). Go to the first server, open Explorer, and navigate to the folder you need to change. Right-click it, and then select Properties. On the Security tab, click Edit.

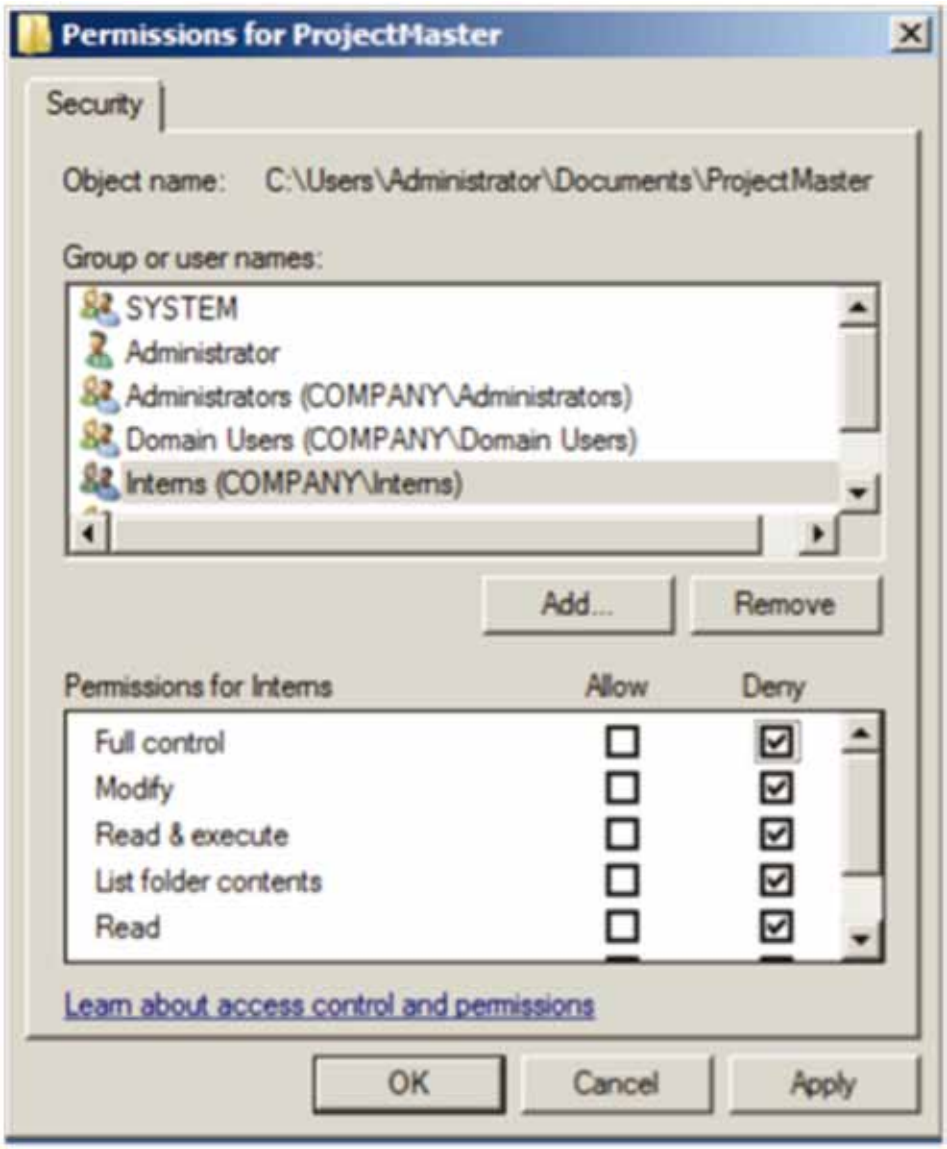

Permissions for ProjectMaster
Security
Object name: C:\Users\Administrator\Documents\ProjectMaster
Group or user names:
SYSTEM
Administrator
Administrators (COMPANY\Administrators)
Domain Users (COMPANY\Domain Users)
Sales (COMPANY\Sales)
Users (COMPANY\Users)
Add...
Remove
Permissions for SYSTEM
Allow
Deny
Full control
Modify
Read & execute
List folder contents
Read
Learn about access control and permissions
OK
Cancel
Apply

Click Add.


Permissions for ProjectMaster
Select Users, Computers, Service Accounts, or Groups
Select this object type:
Users, Groups, or Built-in security principals
Object Types...
From this location:
company.pri
Locations...
Enter the object names to select (examples):
Interns
Check Names
Advanced...
OK
Cancel

Nobody ever really thought ahead to a time when Windows would be running on hundreds of servers within an organization, and when we'd be managing permissions on millions of files and folders.

Type Interns, and hit OK. Click the first checkbox under "Deny," and click OK.



Wasn't that fulfilling? Nine clicks and a little bit of typing. Now, repeat that on every server where that folder and its files exist. Oh, wait – don't forget to go check every file in the folder to make sure none of them have overriding permissions applied to what's inherited from the folder. Shouldn't take more than a few hours. Per server. Enjoy!

The problem: distributed permissions = awkward

This is the problem inherent in Windows' permissions: each resource keeps track of its own access control list (ACL), and there's no central repository for permissions. While the GUI is great for changing one thing at a time, and while command-line utilities can help with bulk management, the fact is that you have to reach out and touch far too many discrete ACLs. Any kind of "bulk permissions change" becomes a manual, tedious nightmare – with plenty of opportunity for mistakes as your bleary eyes and tired fingers lead you to click the wrong checkbox, miss a file, or make some other simple mistake.

This isn't really a "ding" on Windows. After all, this discretionary access control (DAC) model was created back in the 1980s and 1990s, when Windows wasn't going to be more than a departmental file server. Nobody ever really thought ahead to a time when Windows would be running on hundreds of servers within an organization, and when we'd be managing permissions on millions of files and folders.

In fact, Windows' file permissions system is a lot like the systems used in Unix, NetWare, and pretty much every other major network operating system. These DAC-based systems are just hard to manage when you've got a ton of files and folders, that's all.

But this problem is solvable.

Solving the problem

Getting all of the permissions into a single place

The basic approach to fixing this nightmare is to get all of the permissions into a single place. There are different ways to do that. One way is to create a database and periodically copy every resource permission in your environ-ment into that database. Doing so generally involves installing specialized agents on your servers so that permissions from files, Exchange, SharePoint, and so forth can all be captured and maintained over time.

Another similar but less-intrusive approach is to have a tool that can reach out to your various servers from a central console, scanning existing permissions and creating whatever reports or changes you need. Such a tool would need to be highly multi-threaded, so that it could scan through permissions more quickly, and do so across multiple machines at once. The tool would likely cache permissions locally to speed up processing for large operations, such as scanning large folder hierarchies.

With all of the permissions in one place, you can easily create ad-hoc reports like "what does this guy have access to?" and perform bulk changes like removing a given user group from multiple resources. You're really just following the same methodology that you would to perform this task manually, but a well-written, multi-threaded tool can do so thousands of times faster. And, software tools like this are obviously automated, so you don't have to stick around while it works. You can go grab a soda, come back, and start working with your results. A bonus here is if the tool natively understands not only file permissions, but also the other secured resources in your environment, like Exchange and SQL Server and so forth. Then, rather than having to mess with each product's individual console, you can do all of your permissions management from one spot.

Security Explorer

Security Explorer is the permissions management tool with all the bonuses. Using a single, centralized console, you can manage permissions on files, folders, registries, printers, services, tasks, SharePoint, SQL Server, and Exchange. Everything. From one spot.

Security Explorer can find every place where a user has permissions, and generate reports to meet management or compliance requirements. It can take that information and "clone" a user, too, solving the age-old task of "just give the new guy the same permissions as this coworker." In addition to reporting on permissions, Security Explorer can make bulk changes to them, adding groups, removing groups, and so forth.

It can even back up and restore permissions independently of the resources they apply to, enabling you to create and maintain security baselines that represent your organization's security requirements and best practices.
So why not visit www.quest.com/security-explorer and see how Security Explorer can take the pain and anguish out of permissions management and reporting? There's a free trial, along with a product walkthrough and lots more information.

Security Explorer is the permissions management tool with all the bonuses.

About Dell

Dell Inc. (NASDAQ: DELL) listens to customers and delivers worldwide innovative technology, business solutions and services they trust and value. For more information, visit www.dell.com.

If you have any questions regarding your potential use of this material, contact:

Dell Software
5 Polaris Way
Aliso Viejo, CA 92656
www.dell.com
Refer to our Web site for regional and international office information.