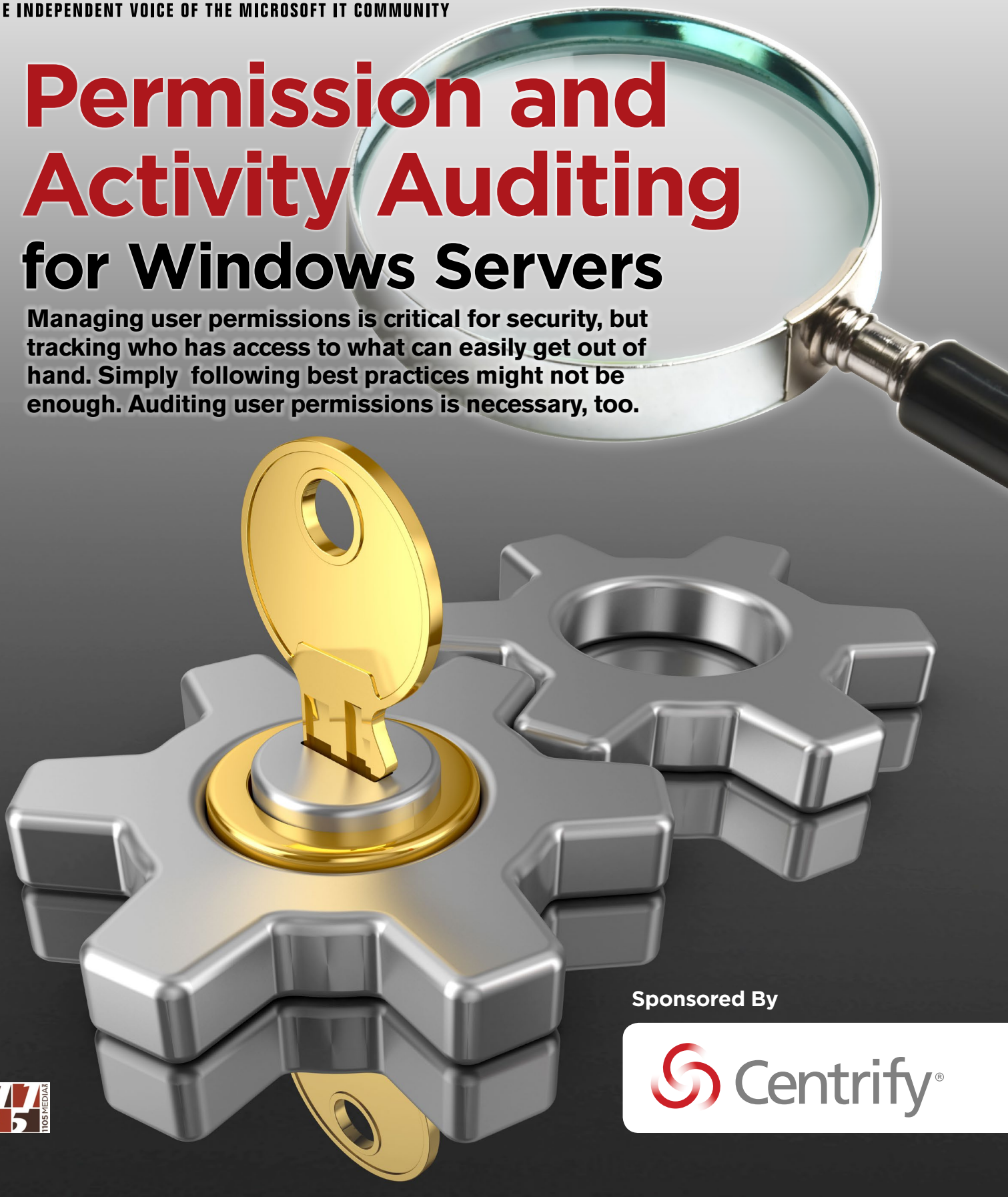
Redmond
THE INDEPENDENT VOICE OF THE MICROSOFT IT COMMUNITY

Permission and Activity Auditing for Windows Servers

Managing user permissions is critical for security, but tracking who has access to what can easily get out of hand. Simply following best practices might not be enough. Auditing user permissions is necessary, too.



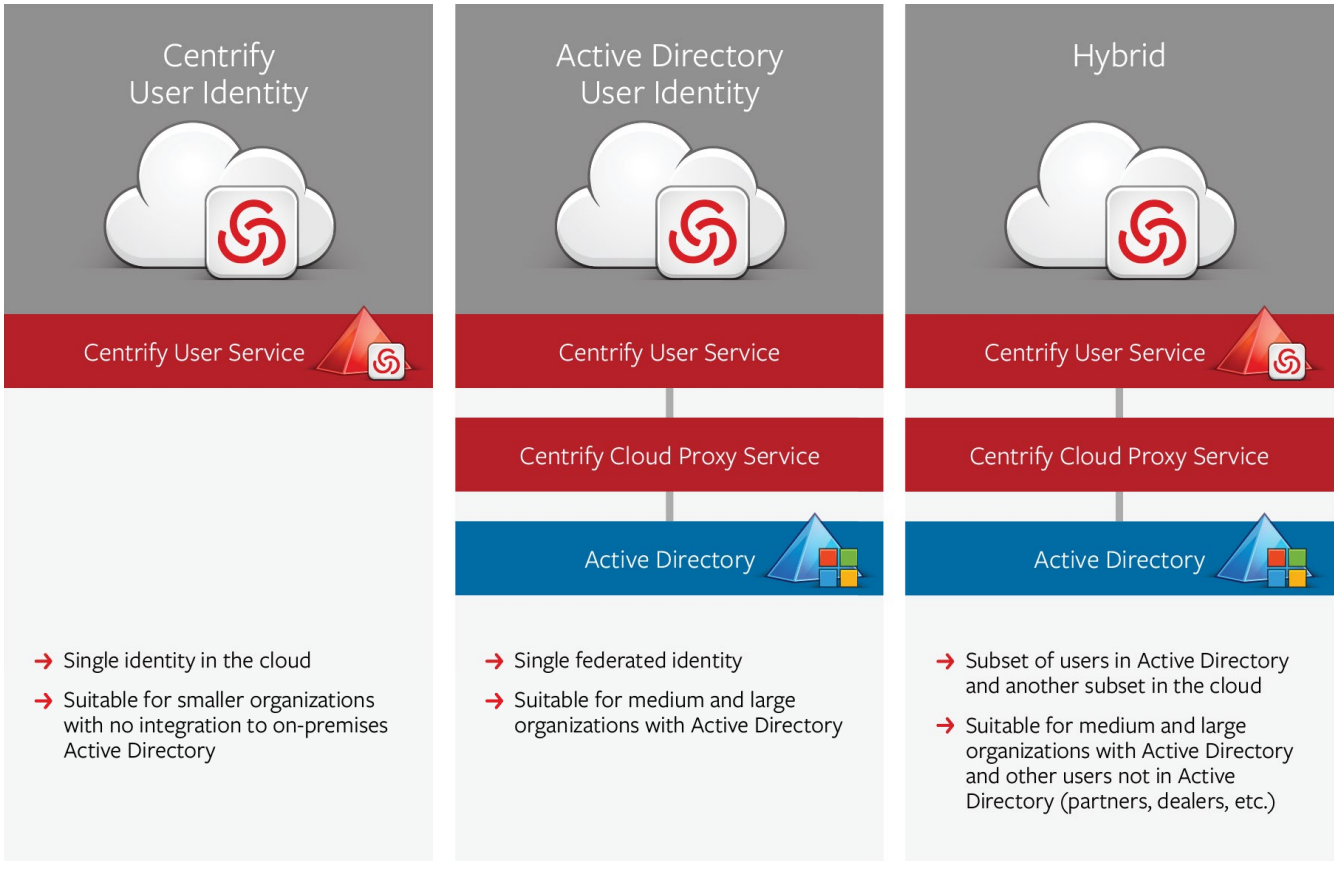
1105 MEDIA

Unified Identity Where You Want It

Centrify provides Unified Identity Services across the data center, cloud and mobile that results in one single login for users and one unified identity infrastructure for IT. Centrify's solutions reduce costs and increase agility and security by leveraging an organization's existing identity infrastructure to enable centralized authentication, access control, privilege management, policy enforcement and compliance.

Because Centrify provides test support and turnkey integration with thousands of SaaS apps, complete mobile app management and on premise plug-ins for SAP NetWeaver, Java and web applications as well as databases, you can get SSO across cloud, mobile and on-premise applications from one single vendor.

Additionally, with Centrify User Service, you need not sacrifice control of your corporate identity. Already considered the innovative leader in leveraging Active Directory, Centrify integrates the Centrify Cloud Service with Active Directory without poking extra holes in your firewall or adding devices in your DMZ. And unlike other solutions, Centrify does not make the fundamental security mistake of duplicating AD into the cloud, maintaining your organization's identity inside Active Directory and under your control.



Why not try it for free today?

www.centrify.com/saas-free-trial

Centrify®

Permission and Activity Auditing for Windows Servers

BY BRIEN M. POSEY

Most IT professionals accept the idea that users should be granted the least possible privileges required to do their jobs. However, experience has shown that users tend to accumulate permissions over time and may retain permissions that are no longer required. As such, adhering to established best practices for the assignment of permissions is not enough. Administrators need a way to periodically compare users' effective permissions against required permissions. Furthermore, it is also important for administrators to determine how a user's permissions are being used.

Although there are a number of native tools that allow Windows administrators to perform permission audits, permission management and usage auditing has always been a tedious and labor-intensive process. This whitepaper explores some possible solutions to the challenge of permissions management.

Important Auditing Considerations

When it comes to auditing, it is best to take a two-tiered approach. The first of these tiers involves auditing user's permissions. The goal behind doing so is to determine what users are theoretically capable of doing based on the permissions that have been granted to them.

The only way to get a comprehensive view of your network security is to examine both permissions and events.

The second tier is event auditing. Event auditing allows an administrator to determine what a user has done or has tried to do. This type of auditing allows you to see when a user is making use of the permissions that have been granted, as well as when security permissions have prevented a user from taking some sort of action.

The only way to get a comprehensive view of your network security (with regard to user rights) is to examine both permissions and events. When using native tools, however, permissions and events must be examined separately.

It is also worth noting that in real world environments, there may be other types of permissions that need to be considered, such as application-level permissions. For the sake of this whitepaper, however, the discussion will focus solely on operating system-level events and permissions.

Best Practices for Permissions Management

Even in an ideal situation, evaluating and auditing user permissions through native tools is challenging. These challenges can be further compounded if administrators do not follow the established best practices for permission assignments.

Microsoft has long stated that users should never be granted direct access to files and folders (with the exception of the user's profile directory). Instead, file and folder permissions should be applied to groups. User permissions can therefore be effectively controlled by adding or removing group memberships.

In theory, this approach should ensure that each user receives the correct permissions – nothing more, nothing less. In reality, however, it is difficult to maintain the correct permission set over the long term.

The reason for this is simple. Managers are quick to ask that a user be granted permissions but often neglect to request that permissions be revoked. Suppose for instance that a user works in the Marketing department and is given access to a network folder named Proposals. Now let's pretend that after a couple of years, that user is promoted to a management position within the marketing department.

Managers are quick to ask that a user be granted permissions but often neglect to request that permissions be revoked.

As a manager, the user no longer needs access to the Proposals folder because creating proposals is not part of their new job. Instead, the user is responsible for bidding on new business. As such, the head of marketing asks that the user be granted access to a network folder called Financials. However, the head of marketing overlooks the fact that the user no longer needs access to the Proposals folder. The end result is that the user ends up with excessive permissions.

Although the user's excessive permissions might seem harmless in this example, the story clearly illustrates how a user's permissions can deviate from the idea that each user should receive the lowest level permissions required in order to do his or her job. Just imagine how many excessive permissions a user might accidentally retain if the user was promoted to a position in a different department.

Evaluating a User's Permissions

The example that was described in the previous section highlights the need for a periodic review of the permissions that have been assigned throughout the organization. However, doing so can be surprisingly difficult.

The Windows operating system lacks a good permission-auditing utility. In previous versions of Windows, Microsoft recommended using the Cacls.exe utility to verify (and modify) permissions. This utility still exists in Windows 8 and Windows Server 2012 but has been deprecated. The preferred utility is Icacls.exe.

Like its predecessor, Icacls.exe is a command line utility that can be used to verify (and when necessary, modify) file and folder

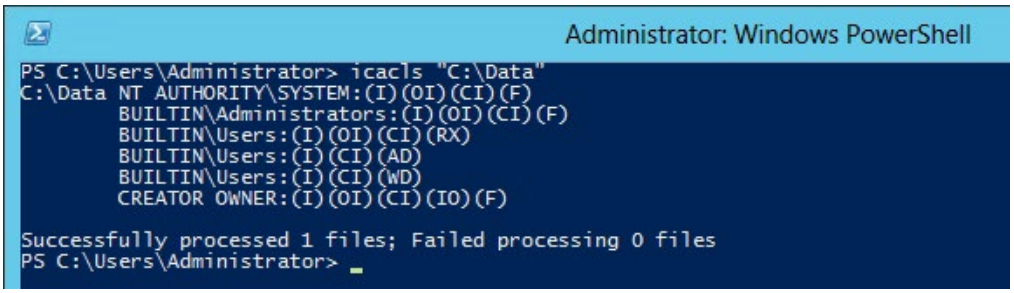


Figure A: *The Icacls.exe command can be used to view or modify permissions.*

Icacls.exe is a command line utility that can be used to verify (and when necessary, modify) file and folder permissions.

permissions. For example, if an administrator wanted to determine the access rights for a folder named C:\Data, then the administrator could use the following command:

```
Icacls "C:\Data"
```

You can see what this command looks like in **Figure A**.

As you can see in the figure above, this command shows which groups have access to the specified folder and what permissions have been granted to each group. Of course, this does not tell us

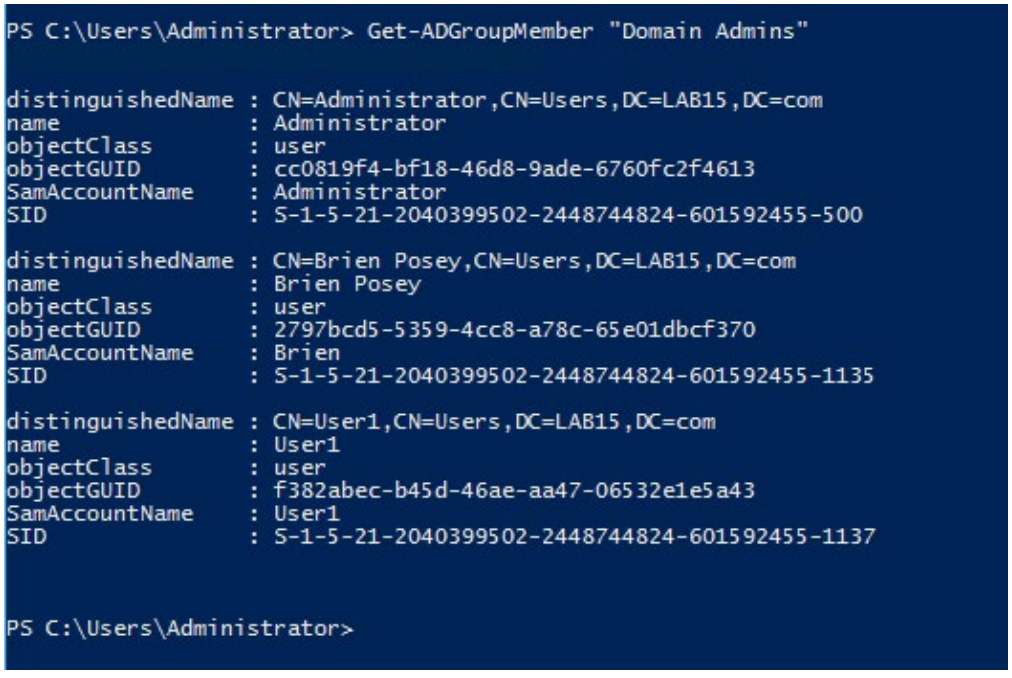


Figure B: *The Get-ADGroupMember cmdlet can be used to retrieve group membership.*

who actually has permission to access the folder. To find that out, we need to take a look at group memberships. The easiest way to accomplish this is through a PowerShell cmdlet called Get-ADGroupMember. Suppose, for example, that you wanted to find out which users belonged to the Domain Admins group. The command used for doing so would be:

```
Get-ADGroupMember "Domain Admins"
```

You can see this cmdlet in action in **Figure B**.

As you look at the figure above, you will notice that one of the members of the Domain Admins group is User1. If you were attempting to conduct a security audit, you might want to know what other groups User1 belongs to. You could determine this information by using the following PowerShell command:

```
Get-ADPrincipalGroupMembership User1 | Select name
```

You can see what this command looks like in **Figure C**.

Although it is possible to retrieve access control information directly from the Active Directory, doing so requires a strong working knowledge of PowerShell.

These cmdlets illustrate the fact that, although it is possible to retrieve access control information directly from the Active Directory, doing so requires a strong working knowledge of PowerShell. Sure, the cmdlets used here were somewhat simple, but in the real world the cmdlets that would be required would tend to be more complex. For one thing, you would probably need to use various filters as a way of narrowing down the results. You would also need to know how to export the permissions data to a report file.

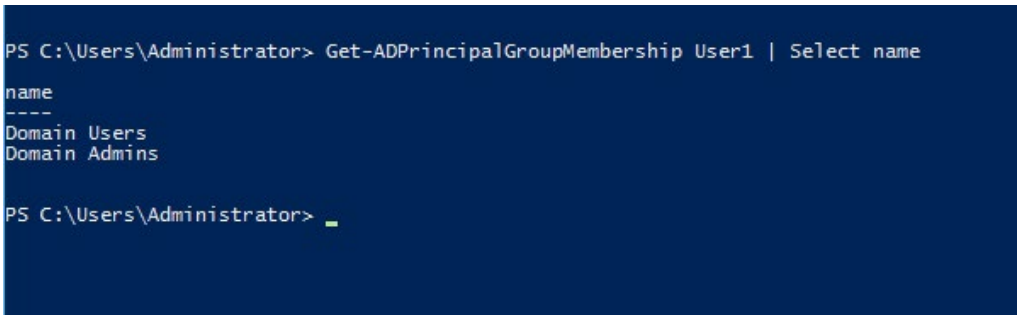


Figure C: *The Get-ADPrincipalGroupMembership cmdlet can be used to retrieve a user's group memberships.*

Of course, this raises the question of whether or not Windows offers any sort of graphical interface for permissions reporting. No such utility is natively included with the Windows operating system, but Microsoft does offer a free utility from Sysinternals called AccessEnum (http://technet.microsoft.com/en-us/sysinternals/bb897332.aspx). The AccessEnum utility is designed to provide you a list of who has read access and who has write access to a folder and all of its sub folders. The utility also shows you who has been denied access to the folder. You can see what the AccessEnum utility looks like in **Figure D**.

As you can see, the AccessEnum utility makes it easy to view file and folder permissions, but it lacks the features of a comprehensive reporting tool.

Evaluating access control permissions is really only part of the auditing process.

Event Auditing

As previously discussed, evaluating access control permissions is really only part of the auditing process. Auditing can only be effective when events are also audited. The Event Viewer has long been the standard tool for confirming policy compliance.

The down side to the native event logging tools is that security event logging is something of a delicate balancing act. If security events are logged at a granular level, then the sheer number of security events that are recorded will typically be so excessive that the logs become meaningless. Conversely, taking a minimalist approach to logging can cause important security events not to be logged.

Microsoft's recommendation has long been that you should only audit the most important events, so as to prevent the security logs from growing to an overwhelming size.

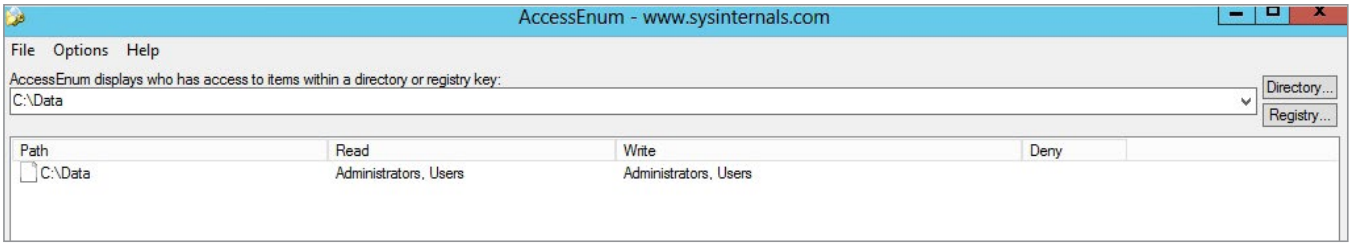


Figure D: *The AccessEnum utility is a graphical tool for viewing access control permissions.*

A Final Word

The Windows operating system contains a variety of tools that administrators can use to audit permissions and events across the organization. Even so, Windows lacks a good reporting engine for security data. This isn't to say that organization-level auditing using native tools is impossible. It's just that an administrator who wants to depend solely on native tools is going to need a bit of creativity and a strong working knowledge of PowerShell. When you consider the time and effort that are required for manual permissions auditing, most organizations are probably going to be much better off investing in third-party tools that can help them to evaluate permissions and policy compliance.

Windows lacks a good reporting engine for security data.

Obviously, every third-party tool is different, but at least some of the tools that are available offer a change tracking feature that allows an administrator to view how permissions have evolved over time. This is something that would be very difficult to accomplish through PowerShell. R

Brien M. Posey is a seven-time Microsoft MVP with more than two decades of IT experience. He's written thousands of articles and several dozen books on a wide variety of IT topics. Visit his Web site at brienposey.com.