Realtime publishers

The Essentials Series: Tackling Active Directory's Four Biggest Challenges

Enabling Useful Active Directory Auditing

sponsored by

QUEST SOFTWARE®

by Greg Shields

Copyright Statement

Enabling Useful Active Directory Auditing

Let's be honest. Auditing in Active Directory (AD) has long been a painful and difficult experience, one that few of us ever enjoy. AD's nine historical auditing categories—account logon events, directory service access, and system events, to name a few—were created for the large-scale dumping of huge amounts of data into a single storage location. Those nine audit categories could be turned on or off, but little else could be done to filter out data before it began clogging the Security event logs on your Domain Controllers.

That is, until the release of Windows Server 2008. With this release, Microsoft has officially recognized the lack of granularity in these nine logging categories and provided a mechanism for dialing down the level of data your domain stores about itself. In Windows Vista and Windows Server 2008, the old *auditpol.exe* command-line tool is made more powerful with the ability to define subcategories to each of the original nine categories.

The result is that the collection of AD event information can now be tailored to store just your information of interest. If your security policy doesn't care about events associated with computer management but is fully interested in the management of users, you can set just that configuration. If you want more detailed tracking of changes to your AD configuration itself, you can configure that as well with the extended *auditpol.exe* tool.

The Native Solution

Recognize first that AD's new log granularity is exposed through the auditpol command line only. The only mechanism currently possible to manage the enabling and disabling of event classes is by creating policies on each and every Domain Controller in your environment. This means running these commands individually on each Domain Controller under management.

The first two commands you'll need to know are *auditpol.exe /list /category* and *auditpol /get /category:<categoryName>*. These two commands list the syntax for each of the original nine categories, the results of which you can enter as *<categoryName>* to learn about the new subcategories. For example, you use the first command to learn about the "object access" category. You can use the second—*auditpol /get /category:"account management"*—to identify its six new subcategories.

Enabling the Account Management category will begin logging changes to different types of objects in AD such as computer accounts, security or distribution groups, and user accounts, among others. But for many organizations, this may simply be too much data that ends up clogging the available space in the security log. To disable the Security Group Management subcategory, use the command *auditpol.exe /set /subcategory:"security group management" /success:disable /failure:disable*. Alternatively, if you wanted to enable success and failure logging of computer account changes, use the command *auditpol.exe /set /subcategory:"computer account management" /success:enable /failure:enable.*

One brand-new subcategory that is useful for identifying changes to the AD configuration itself is Directory Service Changes, which is part of the DS Access category. Directory Service Changes is new in that it now comprehensively logs the actual configuration changes that occur within AD. For example, this subcategory can log which administrator made a change to a user object, when that change occurred, and what the "before" and "after" values were associated with the change. Directory Service Changes logs four major classes of changes by Event ID:

- **Modified objects (Event ID 5136).** These are events associated with changes to objects and will show the previous and current value of the changed attribute.
- **Created objects (Event ID 5137).** These are events associated with newly created objects, showing the object's populated values at the time of creation.
- **Undeleted objects (Event ID 5138).** These are events associated with undeleted objects, including information about where the object was moved.
- **Moved objects (Event ID 5139).** These are events associated with moved objects, which include the previous and new location of the object.

Directory Service Changes is an important tool to fulfill auditing requirements should your organization mandate the collection and storage of information about administrator activities. Be aware, however, that the enabling of this subcategory can dramatically increase the amount of data being stored in your logs. This is the case because each and every change to AD objects is logged, along with before and after values.

Microsoft provides a way to further tailor the types of events that Directory Service Changes monitors. To do so requires the use of the powerful ADSIEDIT console, a tool that requires much caution to use. After launching ADSIEDIT, navigate to the *Schema* naming context and locate the attributes you do not want to monitor. You'll see that there are hundreds, if not thousands, of potential attributes that can be monitored. For each attribute that you do not want to monitor, set its *SearchFlags* attribute to a decimal value of *256*. The attribute will now report NEVER_AUDIT_VALUE in the interface, and Directory Service Changes will ignore it for logging purposes.

The Need for Better Tools

In Windows Server 2008, Microsoft provides this new and useful way to both trim down and bolster your logging potential. Yet you'll notice that none of these new features actually resolves the central problems behind Windows-based event logging.

First is the problem of managing the data itself. Although auditpol provides a mechanism for trimming down the data that is entered into the log, the tools available for looking at that data once in the log remain challenging. Windows Server 2008 enables new functionality with its updated Event Log viewer. However, many administrators still find that functionality challenging to use across multiple Domain Controllers and with large amounts of data. Needed are external tools that enable the consolidation of log data from multiple sources into a single database. Without such a tool, administrators are required to look across each Domain Controller in the environment to get a clear picture of activities on the domain.

Also problematic is the insecure nature of the Event Logs themselves. Any administrator with Domain Admin privileges has the ability to clear any or all Event Logs. This allows a person to cover their tracks in the case of a malicious event. It also fails a key security test required by many types of regulatory compliance. These compliance regulations require a separate structure for the storage and retrieval of log information with security controls that prevent misuse or malicious deletion.

Smart organizations realize the need to consolidate logs across their network environments into single, centralized databases. These databases provide external control over log data while enabling events to be correlated across multiple machines at once. Best-in-class solutions provide ways for security officers and auditors to view this data without sacrificing its security.