How can I find out which process and user is modifying a file?

devblogs.microsoft.com/oldnewthing/20130827-00

August 27, 2013

Raymond Chen

When troubleshooting a problem, you may discover that a file is being modified that shouldn't, and you figure out would be nice if there were some way of finding out which process is modifying the file (so you can get it to stop). Enter the security auditing system. Every securable object has an associated system access control list (SACL) which controls what audit events are raised when a request is made to access the object. You can say, for example, "Log an event in the security event log if somebody tries to open this file for writing but is denied access," or "Log an event in the security event log if somebody successfully creates a new file in this directory." Here's how it works. Let's say that you want to access successful requests from any user to write to a particular file. View the Properties of the file, go to the Security tab, and click Advanced, then go to the Auditing tab and elevate to administrator if necessary. Next, click Add. What happens next depends on what version of Windows you're using, since the interface varies slightly (but the basic idea remains the same). When asked for the security principal, set the Location to the local computer and use the object name *Everyone*. Older vesions of Windows will give you a grid of options. Look for the row corresponding to the operation you want to audit and check the box under *Successful* if you want to audit successful accesses or the box under *Failed* to audit failed accesses. (Or check both to audit both successful and failed accesses.) Repeat for each access you want to audit. In our case, we would check the *Create files / write data* and *Create folders / append data* boxes under the *Successful* column. Newer versions of Windows break the grid up into two questions. The first is whether you want to audit *Success*, *Fail*, or *All* access. In our case, we want to audit *Success*. The next question is what type of access you want to audit, and in our case we would check *Write*. (Or for finer control, click *Show advanced permissions* and check *Create files / write data* and *Create folders / append data*.) OK your way out of all the dialog boxes to save the changes. All right, let's take this thing out for a spin. Open the file in Notepad, make some change, and then Save them. Now open the Event Viewer and go to the Security event log. And… no log entry. That's because I forgot a step: You have to enable object access auditing. Open the Control Panel and look in the *Administrative Tools* folder. From there, you can run the *Local Security Policy* snap-in. If you are a command line nerd, you can run `secpol.msc`. Under *Local Policies*, *Audit Policy* set the *Audit object access* policy to enable global auditing of successful or failed accesses, depending on what you need.

Okay, let’s try it again. Modify the file and save it. Now go back to the security event viewer and you’ll see audit success events in your log. Again, depending on what version of Windows you’re using, the successful audit event will appear differently. For example, older versions of Windows might show

Event Type:	Success Audit
Event Source:	Security
Event Category:	Object Access
Event ID:	567
Date:	…
Time:	…
User:	…
Computer:	…
Description:	

Object Access Attempt:

Object Server:	Security
Handle ID:	208
Object Type:	File
Process ID:	1964
Image File Name:	C:\WINDOWS\system32\notepad.exe
Access Mask:	WriteData (or AddFile) AppendData (or AddSubdirectory or CreatePipeInstance)

whereas newer versions might show

Keywords:	Audit Success
Date and Time:	…
Source:	Microsoft Windows security auditing
Event ID:	4663
Task Category:	File System

An attempt was made to access an object.	
Subject:	
Security ID:	computer\user
Account Name:	user
Account Domain:	computer
Logon ID:	0x27ADB
Object:	
Object Server:	Security
Object Type Name:	File
Object Name:	C:\test.txt
Handle ID:	0x15c
Resource Attributes:	S:AI
Process Information:	
Process ID:	0xdb0
Process Name:	C:\Windows\System32\notepad.exe
Access Request Information:	
Accesses:	WriteData (or AddFile) AppendData (or AddSubdirectory or CreatePipeInstance)
Access Mask:	0x6

Either way, you can see which process obtained write access to the file, running as what user, at what time. Newer versions of Windows include a bit more information in the event log entry to make it easier to find the access request you're looking for as well as chase the access further. (For example, from the Logon ID, you can figure out which logon session modified the file.) This feature has been around since the beginning of Windows NT, but it seems that very few people know about it. Whenver I point it out to people, they say, "Hey, that's cool. How long has that feature been there?"

Now you too can look smart.

Raymond Chen

Follow