

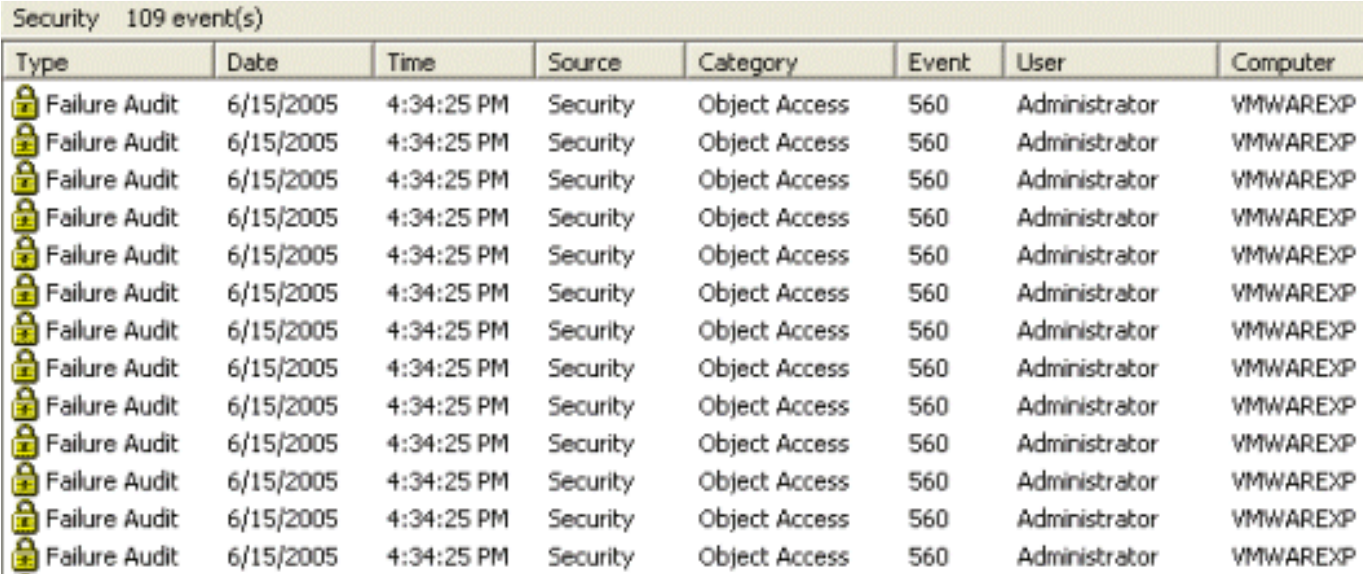
An Explosion of Audit Records













Mark Russinovich
(From Mark Russinovich Blog)

An Explosion of Audit Records

One of the topics I cover in the security module of the Windows internals seminar that I teach with Dave Solomon is auditing. I demonstrate object access auditing by enabling failure auditing in the Local Security Policy Editor (which you launch by typing secpol.msc in the Run dialog), removing all access to a folder on the system, adding failure auditing to the folder's System Access Control List (SACL) in Explorer's permissions editor dialog, and finally, generating failure audits by attempting to navigate into the directory.

In order to verify that audit records generate I open the Security event log in the Event Viewer. Explorer presents an access denied message box exactly once when I try to open the locked-down directory, so when I performed the demonstration for the first time I expected to see one failure audit record. To my amazement and confusion, however, I saw something like this:



Type	Date	Time	Source	Category	Event	User	Computer
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP
 Failure Audit	6/15/2005	4:34:25 PM	Security	Object Access	560	Administrator	VMWAREXP

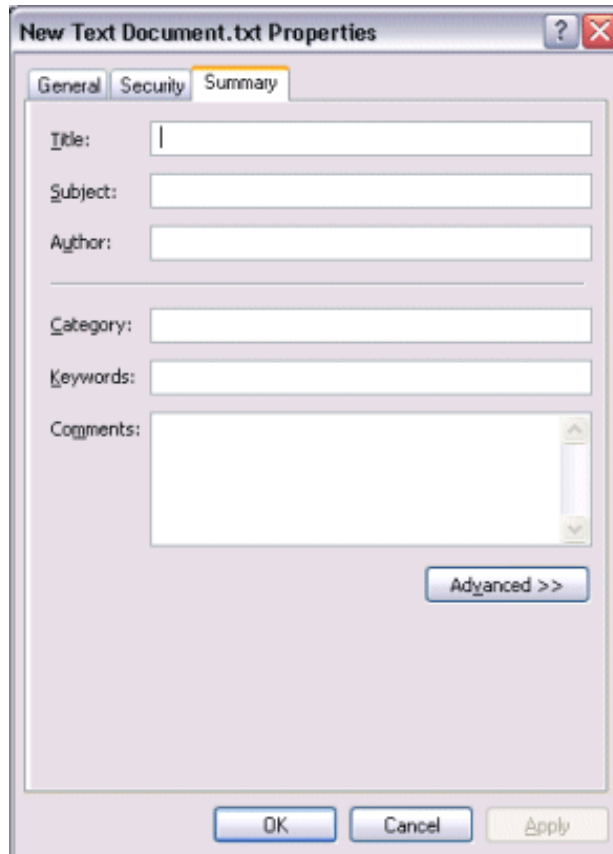
I cleared the security event log immediately before causing the error so the 109 events the Event Viewer reports are in the log are all related to one access denied operation in Explorer. Its no wonder auditing is turned off by default! When one user action generates 109 events it's pretty easy to craft an attack that effectively deletes the useful contents of the security event log.

Every time I've performed the demonstration I've glossed over the fact that there are far more events than I expect to see and make a mental note that I should investigate what's going on. When I finally got around to it my first step was to run Filemon and see what's happening behind the scenes. Filemon confirmed that Explorer is indeed trying to open the file 109 times:

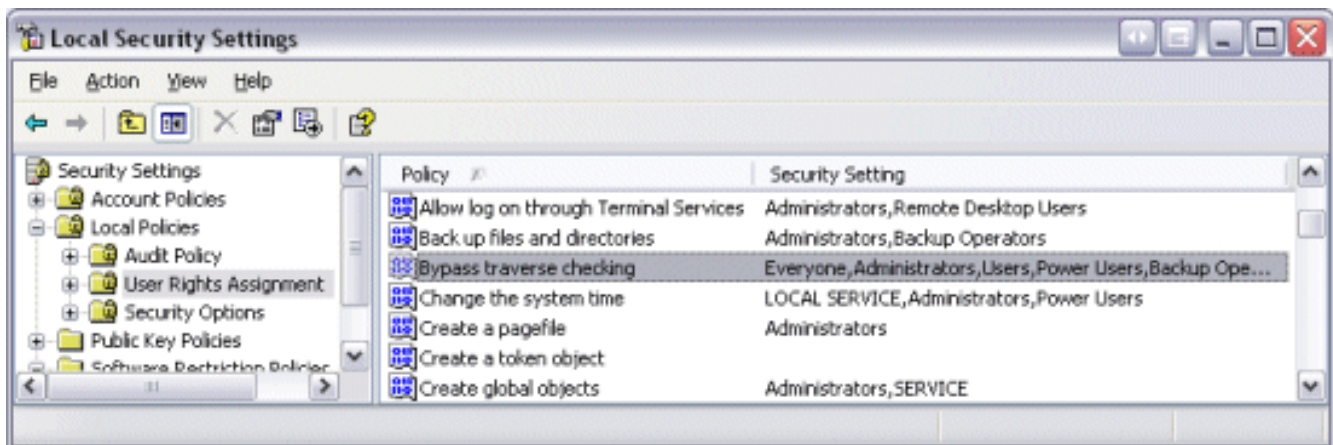
An Explosion of Audit Records

Mark Russinovich
(From Mark Russinovich Blog)

After trying the experiment a few times I realized that the file access attempts occur when I click on the folder, not when I try and navigate into it. It looks like Explorer checks for the presence of several variants of “summary information” alternate data streams on the directory. Not only that, but it looks like it checks for the same ones over and over. These summary information streams are the same ones that Explorer creates when you enter information on the summary tab of a file’s properties dialog box:



If you watch Explorer query a file when you open a file’s properties with Filemon you’ll see the same alternate data stream queries we saw it execute on the locked-down folder:



An Explosion of Audit Records

Mark Russinovich
(From Mark Russinovich Blog)

Several questions remained open to me at this point. First, why is Explorer even looking for properties on a folder when a user simply selects the folder? Second, Explorer doesn't present the Summary tab in the properties dialog box for folders, so why does it expect to find properties on a folder? Finally, why does it try to read the same streams repeatedly, even in the face of errors?

A hypothesis that addresses the first two questions is that the distinction between files and folders with respect to displaying summary information tabs on properties dialog is at a higher layer than the code that checks for the presence of the summary information streams. Besides allowing you to configure columns that show summary information in its Details view, Explorer shows the information in the tooltip it pops open if you hover the mouse over a selected item long enough. It must be attempting to gather that information ahead of time.

I figured a look at Explorer's stack during one of the stream queries might shed some light on the other third question. I couldn't simply set a breakpoint on CreateFile, the API used to open files, because that's one of the most commonly used APIs on a system. Instead, I set the breakpoint on SepOpenAuditAlarm, the kernel-mode Security Reference Monitor function that NTFS calls when it wants to write an event to the security event log. Here's part of the stack trace that I got:

```
ntoskrnl!_IopParseDevice+0A58
ntoskrnl!_ObpLookupObjectName+056A
ntoskrnl!_ObOpenObjectByName+00EB
ntoskrnl!_IopCreateFile+0407
ntoskrnl!_IoCreateFile+008E
ntoskrnl!_NtCreateFile+0030
ntoskrnl!_KiFastCallEntry+00F8
ntdll!_KiFastSystemCallRet
ntdll!_ZwCreateFile+000C
kernel32!_CreateFileW+01B6
shell32!_StgOpenStorageOnFolder+0060
shell32!CFSFolder::AddRef_76339+0041
shell32!CFSFolder::Open+0011
shell32!IsSlowProperty+001C
shell32!CPropStgColumns::GetItemData+00DE
shell32!_IID_IScreenResFixer+045D
shell32!CNameSpaceItemUIProperty::GetPropertyDisplayValue
shell32!CDetailsSectionInfoTask::RunInitRT+017B
shell32!CRunnableTask::Run+004C
browseui!CShellTaskScheduler_ThreadProc+0082
shlwapi!ExecuteWorkItem+001D
ntdll!_RtlpWorkerCallout+0065
ntdll!_RtlpExecuteWorkerRequest+001A
ntdll!_RtlpApcCallout+0011
ntdll!_RtlpExecuteWorkerRequest+00B3
kernel32!_BaseThreadStart+0037
```

The highlighted frame is the one that shows the name of the function, CPropStgColumns::GetItemData, that kicks off the repeated queries. Its name confirms that it has something to do with the Details view columns and likely the tooltip, but the trace doesn't explain why its querying the same streams repeatedly or why it keeps trying when the access denied errors should cause it to give up.