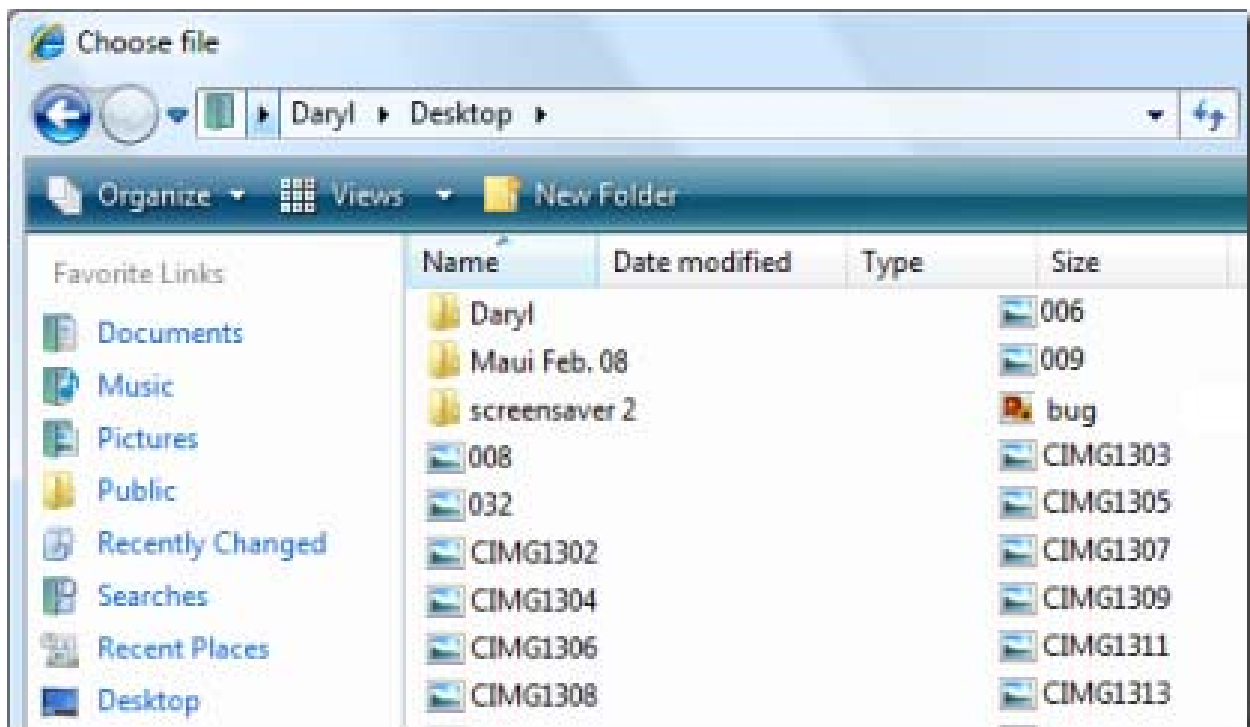


# The Case of the Desktop Files

Mark Russinovich  
(From Mark Russinovich Blog)

A few weeks ago, my wife mentioned that she sometimes saw files in her desktop folder that didn't appear on the actual desktop. She brought it up not only because she was confused by the discrepancy, but because she wanted to move some of these phantom files to other folders and wanted to delete others. I had no idea what she was talking about (which was usually the case when she described her computer troubles), so I told her that the next time she saw these mysterious files, to call me to look at it.

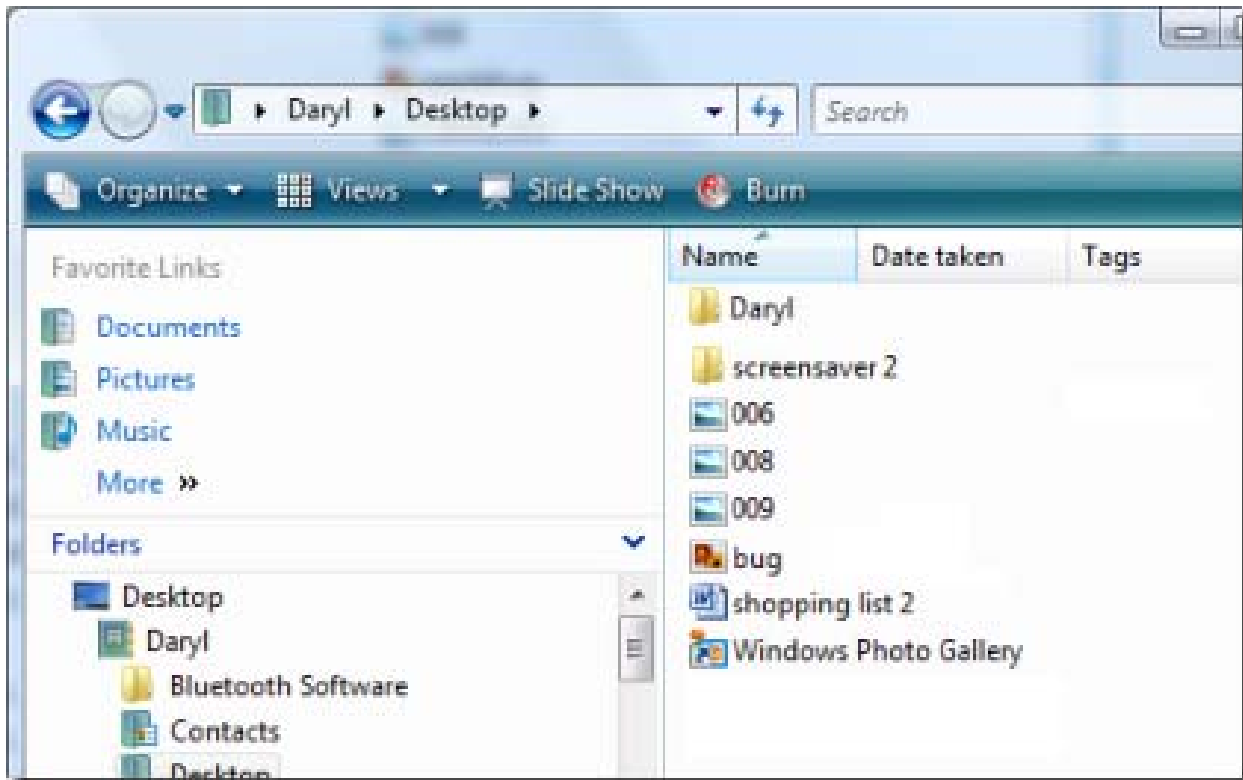
A few days later I got home from work and she greeted me excitedly at the door and explained that the problem reoccurred and that she had left a window open showing the elusive files. I rushed to the kitchen computer with anticipation, not even bothering to greet the dogs on the way, and surveyed the situation. She had a maximized IE window open with a full row of tabs for her open emails (I don't think she ever closes an email window). An IE "Choose a File" dialog box was in the foreground listing the files in her desktop folder, which she had opened by clicking the attachment button in the email editor. The dialog looked like this:



I minimized IE to view the desktop background and sure enough, several of the files visible in the dialog, such as the "Maui Feb. 08" folder and the CIMG13xx JPG files, were missing. I opened an Explorer window and navigated to her desktop folder to see if the files would show up there, but they were missing there as well:

# The Case of the Desktop Files

Mark Russinovich  
(From Mark Russinovich Blog)



I'd never seen that behavior before. I knew this was a job for Process Monitor. Since my wife doesn't keep the Sysinternals tools on her system (sad, but true), I ran it directly from the network using the Sysinternals Live address, `\\live.sysinternals.com/tools/procmon.exe`. With Process Monitor recording activity, I closed and reopened the Choose File dialog from the email editor and then I search for "CIMG", a portion of the file name for many of the files present in the Choose File dialog, but not in the Explorer view of the desktop. The first hit was a directory enumeration operation with the file names showing as results in the Details column on the far right:



The files were located in her profile under `\Appdata\Local\Microsoft\Windows\Temporary Internet Files\Virtualized\C\Users\Daryl\Desktop`. This Virtualized is directory created by IE7 when run in Protected Mode (PMIE), which is the default mode on Windows Vista and Windows Server 2008.

PMIE uses Integrity Levels, introduced in Vista and Server 2008, to limit the file system and registry locations to which code running in IE can modify to a subset of those writeable by the user account in which IE executes. As I described in an earlier blog post, the sandbox defined by locations labeled with Low Integrity, the level at which PMIE executes and of the objects that PMIE can modify, allow PMIE to save favorites and temporary files, like the IE cache and browsing history. However, PMIE cannot modify other locations in a user's account, like documents folders and per-user autostart locations in the registry and file system, because they have an integrity level of Medium. That

## The Case of the Desktop Files

Mark Russinovich  
(From Mark Russinovich Blog)

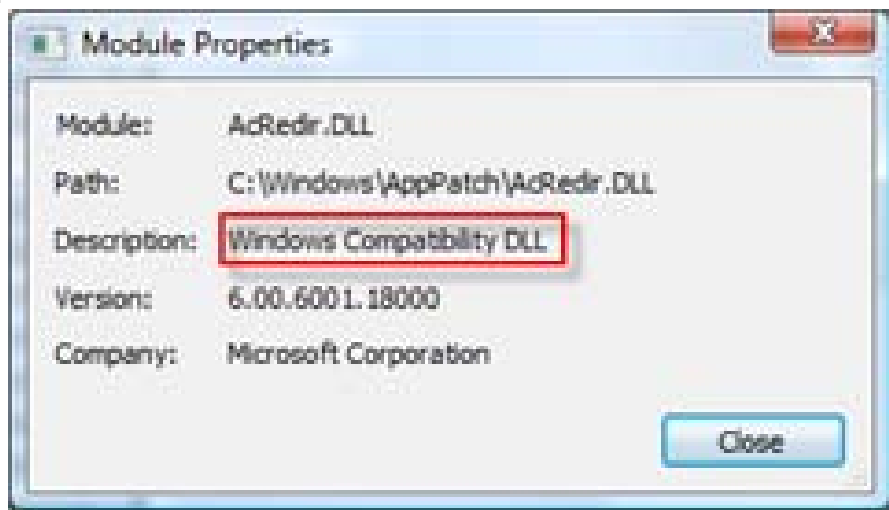
prevents drive-by-download malware that might infect the IE process from establishing a persistent presence.

In order to preserve backward compatibility with legacy code, such as ActiveX controls and Browser Helper Objects, that might be coded to write to locations outside of the sandbox, PMIE implements shims that intercepts file and registry operations and redirects ones that got outside the sandbox to the Virtualized directory within it.

To see if that was what was happening here, I examined the stack trace of the virtualized operation highlighted above by right-clicking on the line and choosing Stack. The stack showed that Acredir.dll was intercepting the operation and executing redirection functions:

U 8	ntdll.dll	ZwQueryAttributesFile + 0xc
U 9	kernel32.dll	GetFileAttributesW + 0x5a
U 10	AcRedir.DLL	NS_RedirectFiles::RedirectorFiles::InitCore + 0x92
U 11	AcRedir.DLL	NS_RedirectFiles::RedirectorFiles::Init + 0x2a
U 12	AcRedir.DLL	NS_RedirectFiles::APIHook_GetFileAttributesW + 0x50
U 13	SHLWAPI.dll	PathFileExistsW + 0x24
U 14	SHELL32.dll	CUsersFilesFolder::_EnumFolders + 0x11a
U 15	SHELL32.dll	CUsersFilesFolder::EnumObjects + 0x31
U 16	SHELL32.dll	CRegFolder::EnumObjects + 0x66
U 17	SHLWAPI.dll	IShellFolder_EnumObjects + 0x68
U 18	browseui.dll	CACLIShellFolder::_SetLocation + 0x74

Double-clicking on the line in the stack trace opens the module properties dialog, which shows that the DLL is the “Windows Compatibility DLL”, thus confirming this was part of PMIE’s sandbox implementation:



# The Case of the Desktop Files

Mark Russinovich  
(From Mark Russinovich Blog)

I had been familiar with PMIE's virtualization, but I'd never seen files virtualized on the desktop, so it had not been obvious to me that was what was causing the discrepancy. Process Monitor revealed the cause, so now all I was left with was cleaning up the virtualized files. Most users don't realize that you can move and delete files from within a file browse dialog, so I took the opportunity to show my wife how she can manage virtualized files from the email editor's attachment dialog if she came across them again. We deleted the files she didn't want and moved the pictures out to her photo library folders.

The case was closed. As a bonus, my wife was impressed at the ease with which I'd figured out the source of the phantom files and even more impressed that I wrote the tool I used to solve it. She'd also gotten an in depth look at PMIE's virtualization and integrity levels, but I think in the end my lecturing on those subjects actually subtracted points.

Incidentally, you'll almost certainly see files and directories if you look at the PMIE Virtualized folder in your profile, because even routine operations within IE result in redirection. Here you can see thumbnail cache files that the shell's file browsing dialog creates when you use it from within IE. Normally, the shell stores thumbnail cache files in your profile, but PMIE can't write to that location so the shim virtualizes it:

