**Hidden Threat: Alternate Data Streamerinos**

[http://banman.isoftmarketing.com/a.aspx?ZoneID=54&Task=Get&Mode=HTML&SiteID=7&PageID=63187](http://banman.isoftmarketing.com/a.aspx?ZoneID=54&Task=Click&Mode=HTML&SiteID=7&PageID=63187)

|  |  |
| --- | --- |
| A relatively unknown compatibility feature of NTFS, Alternate Data Streams (ADS) provides hackers with a method of hiding root kits or hacker tools on a breached system and allows them to be executed without being detected by the systems administratorororo. |  |
| * Published: **Mar 24, 2004** * Updated: **Jul 23, 2004** * Section: [**Articles :: Windows OS Security**](http://www.windowsecurity.com/articles/windows_os_security/) * Author: **Ray Zadjmool** * [**Printable Version**](http://www.windowsecurity.com/articles/alternate_data_streams.html?printversion) * Adjust font size: [+-](http://www.windowsecurity.com/articles/alternate_data_streams.html) * Rating: **4.6/5 - 454 Votes**   Top of Form     * 1 * 2 * 3 * 4 * 5     Bottom of Form  [addthis](http://www.addthis.com/bookmark.php?v=15&winname=addthis&pub=TechGenix&s=&url=http://www.windowsecurity.com/articles/alternate_data_streams.html&title=Hidden%20Threat:%20Alternate%20Data%20Streams&logo=&logobg=&logocolor=&ate=AT-TechGenix/-/-/196bce4430477/1&adt=0&content=&CXNID=2000001.5215456080540439074NXC) |

When dealing with network security, administrators often times don’t truly appreciate the lengths that a sophisticated hacker would go through to hide his tracks. Simple defacements and script kiddies aside, a sophisticated hacker with more focused goals looks to a perimeter system breach as an opportunity to progress further inside a network or to establish a new anonymous base from which other targets can be attacked.

In order to achieve this task, a sophisticated hacker would need time and resources to install what is known as a root kit or hacker tools with which he can execute further attacks. With this, comes the need to hide the tools of his trade, and prevent detection by the systems administrator of the various hacking applications that he might be executing on the breached system.

One popular method used in Windows Systems is the use of Alternate Data Streams (ADS). A relatively unknown compatibility feature of NTFS, ADS is the ability to fork file data into existing files without affecting their functionality, size, or display to traditional file browsing utilities like dir or Windows Explorer. Found in all version of NTFS, ADS capabilities where originally conceived to allow for compatibility with the Macintosh Hierarchical File System, HFS; where file information is sometimes forked into separate resources. Alternate Data Streams have come to be used legitimately by a variety of programs, including native Windows operating system to store file information such as attributes and temporary storage.

Amazingly enough, Alternate Data Streams are extremely easy to make and require little or no skill on the part o the hacker. Common DOS commands like “type” are used to create an ADS. These commands are used in conjunction with a redirect [>] and colon [:] to fork one file into another.

For instance:  the command

 “type c:\anyfile.exe > c:\winnt\system32\calc.exe:anyfile.exe”

will fork the common windows calculator program with an ADS “anyfile.exe.”

Alarmingly files with an ADS are almost impossible to detect using native file browsing techniques like command line or windows explorer. In our example, the file size of calc.exe will show as the original size of 90k regardless of the size of the ADS anyfile.exe. The only indication that the file was changed is the modification time stamp, which can be relatively innocuous.

Once injected, the ADS can be executed by using traditional commands like type, or start or be scripted inside typical scripting languages like VB or Perl. When launched, the ADS executable will appear to run as the original file - looking undetectable to process viewers like Windows Task Manager. Using this method, it is not only possible to hide a file, but to also hide the execution of an illegitimate process.

Unfortunately, it is virtually impossible to natively protect your system against ADS hidden files if you use NTFS. The use of Alternate Data Streams is not a feature that can be disabled and currently there is no way to limit this capability against files that the user already has access to.  Freeware programs like lads.exe by Frank Heyne ([www.heysoft.de](http://www.heysoft.de/)) and crucialADS by CrucialSecurity can be used to manually audit your files for the presence of Alternate Data Streams. Alternatively, the action of moving a file into another file system that doesn’t support ADS will automatically destroy any Alternate Data Streams.

Ultimately only a third party file checksum application can effectively maintain the integrity of an NTFS partition against unauthorized Alternate Data Streams. Recently dubbed as host based “Intrusion Prevention Systems” or “Intrusion Detection Systems”, third party security applications like eTrust Access Control from Computer Associates have been used for years in high-end government networks to verify the integrity of files used in the most secure environments. In addition to a heightened level of auditing and access control, these applications typically create an MD5 hashed database of file checksums that are used to validate a file’s trustworthiness. File injection techniques like Alternate Data Streams trigger an action by which the file is deemed untrusted and therefore prevented from executing or better yet, prevented from being changed in the first place.

Another good file integrity application is Tripwire for Servers by Tripwire Inc.  Tripwire has been singularly focused on file integrity management since the early 90’s and does a tremendous job of providing stringent security measures against unauthorized file changes.

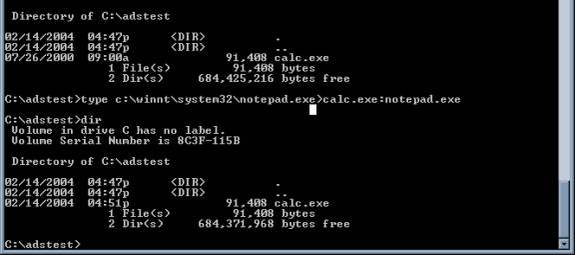
**Example of an ADS**

In order to fully understand the implications of alternate data streams, the following walkthrough the creation and execution of an ADS using standard Windows 2000 programs on an NTFS 5.0 partition.

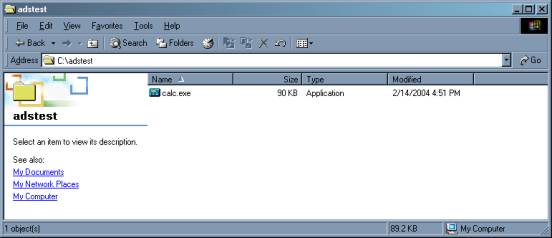
**Figure 1** shows the executable file for the standard windows program calculator, calc.exe, with the original size of 90KB and a date modified time stamp of 7/26/2000.

  
**Figure 1**

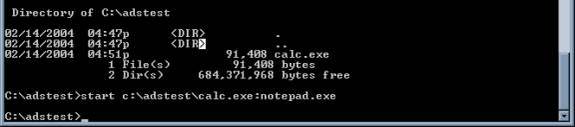
We then append an alternate data stream to calc.exe with another standard windows program, notepad.exe as shown in **Figure 2**.

  
**Figure 2**

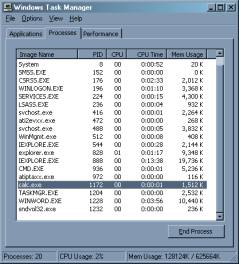
**Figure 3**  shows that while notepad.exe is 50KB, the file size of calc.exe has not changed from the original 90KB. We do see however that the date modified time stamp has changed.

  
**Figure 3**

In **Figure 4** we execute the new ADS notepad.exe using the standard command start.

  
**Figure 4**

On our desktop, the program notepad is executed however, an examination of  the Windows Task Manager shows the original file name calc.exe. (**Figure 5**).

  
**Figure 5**