THE "WOW-EFFECT"

30.11.2011

Author: Christian Wojner (wojner@cert.at)

Summary

The 64-bit version of Microsoft[1] Windows includes file-system virtualization features to run 32-bit programs. File access is transparently redirected to other directories in certain cases.

This feature can easily fool an analyst looking at a running system and can have a massive impact on infection-driven forensics, malware analysis and comparable investigations.

In the worst case this can lead to an entirely wrong interpretation of a case/situation.

While this issue is not entirely new, it is necessary to raise the IT-Security community's awareness, as some of the common tools and procedures in use need to be adapted in the presence of the files system redirector.

Content

[1] I worked with Microsoft on the following analysis and they are aware of my research.

Background

64-bit versions of Windows provide backward compatibility for 32-bit executables. The WOW64 ("Windows-32-bit-on-Windows-64-bit") subsystem provides a 32-bit environment for 32-bit processes including 32-bit versions of various 64-bit executables (PE-Files like .EXE, .DLL, …). Both accesses to the registry as well as to the file-system are transparently intercepted and redirected.

The goal was to give a 32-bit process the impression of running under a native 32-bit Windows operating system instead of the reality – a 64-bit Windows operating system.

In the following, I will focus on how WOW64 modifies access to `%windir%\System32` (typically `C:\Windows\System32`, shortened to `System32` in the rest of this document, just as SysWOW64 for `%windir%\SysWOW64`).

The WOW64 File System Redirector and System32

Microsoft describes this aspect of WOW64 in these words[2]:

> *The %windir%\System32 directory is reserved for 64-bit applications. Most DLL file names were not changed when 64-bit versions of the DLLs were created, so 32-bit versions of the DLLs are stored in a different directory. WOW64 hides this difference using a file system redirector.*
>
> *In most cases, whenever a 32-bit application attempts to access %windir%\System32, the access is redirected to %windir%\SysWOW64. Access to %windir%\lastgood\system32 is redirected to %windir%\lastgood\SysWOW64. Access to %windir%\regedit.exe is redirected to %windir%\SysWOW64\regedit.exe.*

In other words, a 32-bit application will never see any file stored in `System32`, it will always access `SysWOW64` instead. If you need more details on WOW64 feel free to visit the links provided at the end of this report.

Impact – The "WOW-Effect"

While this redirection solution might be optimal for typical customer it is a massive pitfall for IT-Security. The problem is **not the redirection itself**. It's the **way it's done**. Instead of being restrictive to specific situations where i.e. an OpenLibrary or CreateProcess happens or anything else that definitely tries to functionally use "flavored" (32/64-bit) code, this redirection is done on a much lower level than it might have been necessary[3]. In fact even if the 32-bit executable just wants to read the content of PE-File in `System32` **on purpose** it will be redirected to SysWOW64 dealing with a therefore **different** file.

Unfortunately all this redirection functionality happens without any indication.

[2] http://msdn.microsoft.com/en-us/library/aa384187%28VS.85%29.aspx
[3] I assume Microsoft had their reasons.

The impact is that if you are using a 32-bit program (or tool) to deal with specific files in System32 you will access different ones. This also applies to iterating through the directory System32 with FindFirstFile/FindNextFile.

Practical examples

Checking file hashes

Let's assume you are analyzing malicious behavior on a 64-bit Microsoft Windows machine and the malicious activity seems to originate from an executable under System32. The next step might be to run your md5sum tool (which is probably a native 32-bit executable) on it to check it against the hashes of known good files. Chances are high that nothing suspicious can be found that way because of the WOW64 redirection caused the wrong file to be hashed.

Let's try this on my laptop running Microsoft Windows 7 Professional 64 Bit (SP1). Assume the malicious file is the 64-bit DLL file `ieapfltr.dll` under System32. At the time of writing the correct MD5 hashes of the 32- and 64-bit versions of this file are:

File	Type	MD5 Hash
`C:\Windows\system32\ieapfltr.dll`	`64-bit`	`8eada158d964e3fd1999ad96c9c507ff`
`C:\Windows\SysWOW64\ieapfltr.dll`	`32-bit`	`ee9d715af1b928982f417238b9914484`

Running a 32-bit md5sum tool on `C:\Windows\system32\ieapfltr.dll` shows

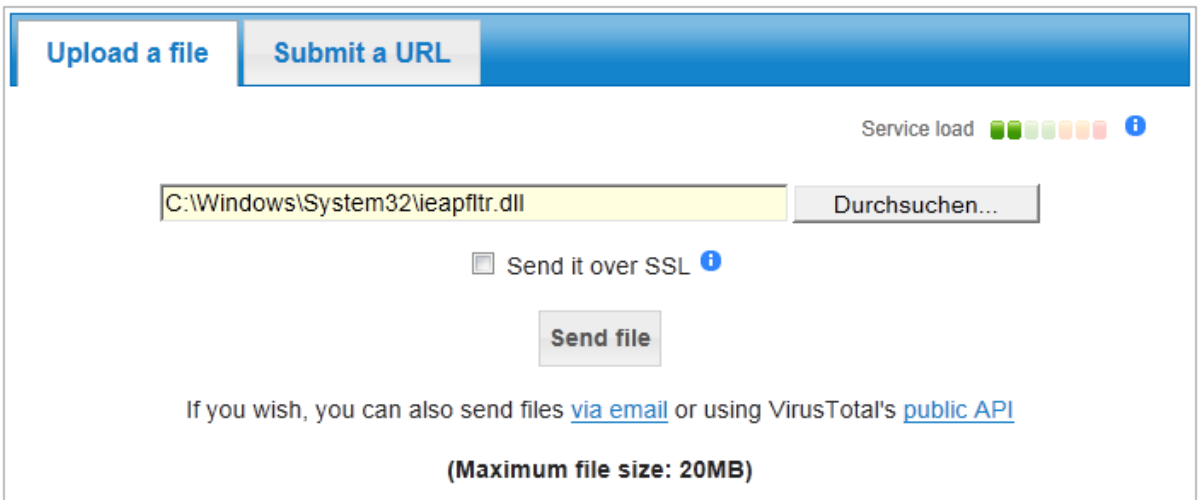
```
F:\>md5sum.exe c:\Windows\System32\ieapfltr.dll
ee9d715af1b928982f417238b9914484  c:\Windows\System32\ieapfltr.dll
```

This is what I call a "WOW-Effect"!

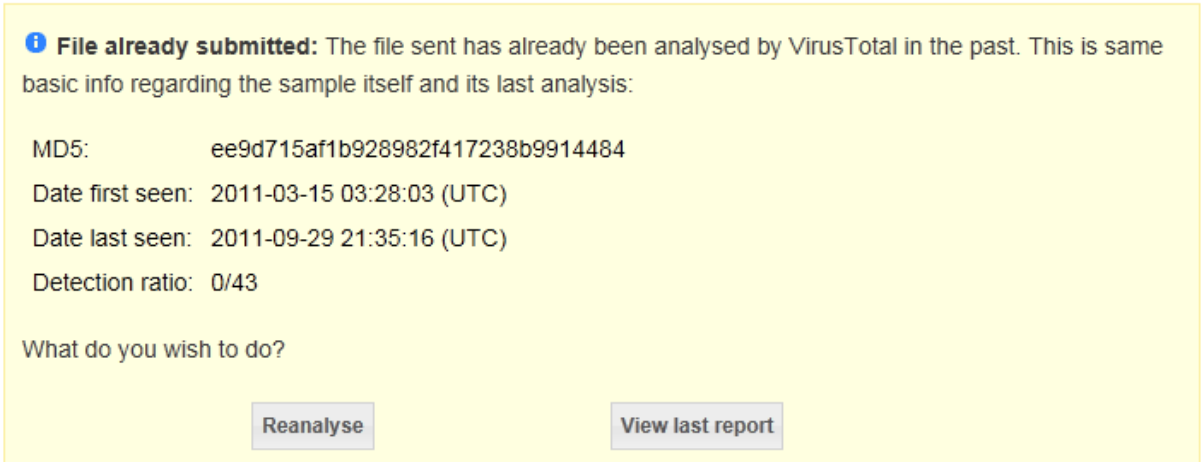
We are not dealing with a bug or vulnerability. It is a feature that can have a massive impact on (typical) IT-Security relevant investigations.

Uploading a malicious file to VirusTotal

A common approach to suspicious files is to upload them to a webservice which checks submitted files against a battery of AV software. This is usually done with the default browser, for this example we use VirusTotal.com:



Take a look at the following screenshot and compare the shown MD5 hash with the ones in our initial md5sum example:



By the time of writing this was true for (at least) the current versions of Firefox, Opera and Internet Explorer. If are you wondering about Internet Explorer: Windows 7 includes both a 32- and a 64-bit version, the default seems to be the 32-bit executable. You can check which version you are running by opening the task manager. 32-bit executables are flagged with a "*32" there.

Iterating

Sometimes researchers write short programs to analyze all files on a computer. This could be a signature scan for malware, just looking for encrypted files, diffing tools, or anything else that iterates over the whole file-system.

As the FindFirstFile call for `System32` will be redirected to `SysWOW64`, the files in the real System32 will not be seen. In other words, if a 64-bit malware is dropped as a unique executable in System32 it is well hidden from any 32-bit tool.

Note: You cannot circumvent the "WOW-Effect" by changing directly into `System32` by running your tool from there, even if you copied your tool right into `System32`.

Conclusions

Regarding the WOW-Effect the most important question for you is whether your tools are still 32-bit only or include 64-bit versions. This is something that might not have been of interest in the past.

Note: It's widely known that nearly all binaries of a clean Windows install have already been uploaded to VirusTotal in the past. Thus I find it interesting that most of the real 64-bit versions I tried were not known at VirusTotal. This indicates that people have not been uploading the right files.

Is Anti-Virus software also affected?

The impact of WOW64 on AV software is not trivial to handle[4]. As AV needs to run at very low level – including kernel modules – it needs to be of the same "flavor" as the underlying operating system. In the case of 64-bit Windows it has to cope both with 32 and 64 bit processes.

Malware has been predominantly 32-bit so far, how AV software handles the transition to 64-bit will be worth watching.

What about my tool set?

Do I have all the tools in 64 bit versions? Alternatively, are the 32 bit versions dealing with the issue correctly, i.e. are they turning off the WOW64 translations before opening files[5]?

What about the past?

Have you done any investigations on a Microsoft Windows 64-bit operating system in the past and could not find anything evil though there were strange symptoms initially?

What Now?

There are much more situations and scenarios where the "WOW-Effect" can have an impact on investigations. I bet you can instantly come up with a lot more of potential scenarios.

Suggestions for IT-Security analysts

- Use 64-bit tools on 64-bit Windows operating systems where you can and you should not experience any unexpected side-affects.
- If you need to write your own tools, e.g. a scanner for a certain type of infection, you need to consider the WOW-Effect.

Final remarks

- The 64-bit DLLs are now in `System32`, the 32-bit versions in `SysWOW64`. Wouldn't it have been smarter to invent a new `System64` for the first 64-bit version of Windows?
- We are all wondering how this will work out for any upcoming 128-bit version of Windows: `System32` has 128-bit, `SysWOW128` 64-bit, and `SysWOW64` contains the 32-bit versions?

[4] http://www.aavar.org/avar2008/abstract/WOW64_Woes_for_Anti-Virus_Products.htm
[5] http://msdn.microsoft.com/en-us/library/aa365743%28v=VS.85%29.aspx

Links

- http://msdn.microsoft.com/de-de/library/aa384274.aspx
- http://msdn.microsoft.com/en-us/library/aa384187%28VS.85%29.aspx
- http://en.wikipedia.org/wiki/WoW64
- http://blogs.sepago.de/helge/2008/04/20/windows-x64-all-the-same-yet-very-different-part-7/
- http://blogs.sepago.de/nicholas/2009/08/26/what-does-wow64-mean-for-application-management/
- http://www.samlogic.net/articles/32-64-bit-windows-folder-x86-syswow64.htm
- https://www.techsupportalert.com/content/how-windows7-vista64-support-32bit-applications.htm
- http://www.aavar.org/avar2008/abstract/WOW64_Woes_for_Anti-Virus_Products.htm