PSBits / IFilter /

Name	Last commit message	Last commit date
..		
Dll.cpp		
FilterBase.h		
FilterSample.cpp		
FilterSample.def		
FilterSample.dll		
README.md		

README.md

Windows Search allows users to perform queries about files metadata (such as name) and about the content. Obviously, files content cannot be always indexed as a plaintext. For example, Word document is a kind of an archive, so it must be opened, read etc. to be properly indexed. Same applies to the PDF documents and many others. To provide a universal (means: adjustable to any file) method, Microsoft created an interface called IFilter. The interface is documented and well known for years. After installing the proper IFilter DLL (usually provided by the vendor responsible for the file format), the Indexing Engine (such as WSearch service in Windows 10) loads it when the content of a particular file type must

be understood. Let's say it again: service loads a DLL, when the file appears and/or changes. We have couple of beautiful circumstances to be taken under the consideration:

1. The code is distributed as DLL, and not EXE, which lowers the visibility.
2. DLL is loaded by the service running as LOCALSYSTEM, which means its code is quite powerful.
3. DLL is not present in memory until the file needs to be indexed. It makes it harder to detect.
4. IFilters are not covered by Autoruns from Sysinternals, being usually used for detection of such potentially malicious (or just unknown) DLLs.
5. IFilters are loaded automatically when a file with particular extension appears. It means the code may be brought to the life after receiving a file as an attachment, downloading it as a part of the webpage etc.

If we combine all these facts, we can create an IFilter, which:

1. Registers itself as the handler for the special file extension (i.e. .attack).
2. Waits for the activation (meaning such file arriving to the machine), possibly months later.
3. Reads the file content.
4. Behaves accordingly to the file content using the LOCALSYSTEM context.

Effectively it means IFilters are quite powerful way of persistence with some remotely initiated/defined capabilities.

It is not a bug, as installing the IFilter requires administrative privileges to create registry hives under the HKLM\SOFTWARE\Classes\CLSID, but at the same time it opens new (at least for me) opportunities.

I am providing simple and innocent PoC DLL. It reacts on .filtersample extension, and reads the file content. The activity may be observed using dbgview.exe from SysInternals. Register/unregister via regsvr32.exe.

The source code is not beatiful, but it works. Feel free to contribute.