**Process Hollowing – a Research**

**Preview**Process Hollowing is process injection sub-technique, where the attacker replaces the legitimate code the was mapped to the memory from a running process, with his malicious code, and by this way, the malicious code will run in the same permissions as the legitimate code that was created in the beginning.

Attacker will use process hollowing when he will want to disguise his malicious code, without the risk that he will damage the activity of the legitimate running process and make the user to suspect the something is not right on the computer.

Process Hollowing is used as AV Evasion technique, that the attacker will use, to avoid detection by AV engines.  
One of the most famous worms, which used Process Hollowing, was Stuxnet in 2010.

**Implementation of the Attack:**In process Hollowing the malware creates new instance of legitimate process in suspended mode, for example svchost.exe.  
After the creation of the process in suspended mode, the malware unmapped the legitimate code from the memory and instead the legitimate code, mapped the malicious code into the memory in the same place, and resume the process, so he will start running as svchost.exe.  
The steps malware takes, to perform Process Hollowing (According to "Mastering Malware Analysis" - Stuxnet secret technique-process hollowing [163-165]):

1. First, the malware creates a legitimate process in suspended mode.  
   By creating the process in suspended mode, the process is created, but his first thread is not running.
2. Hollowing out the legitimate process, by using VirtualFreeEx API for example.
3. Allocate the same space in memory as the unloaded PE image for the malware PE image.
4. Inject the malware executable into the freed memory space by loading the PE file and fixing its import table.
5. Change the thread starting point to the malware entry point.
6. Resume the suspended thread to execute the malware from the entry point.

**Note:**There are many ways to implement Process Hollowing, and this is just an example of one way to implement this technique.

**Process Internals**In order to understand the next parts it is important to be familiar with process internals and with two important terms: PEB and VAD.

Each windows process is represented by an executive process structure called EPROCESS.  
EPROCESS contains attributes that are related to the process and points to other data structures.  
Most of the data structures that EPROCESS points to exists in the kernel-mode, except from one data structure – Process Environment Block (PEB).  
  
PEB is user-mode data structure, and the reason why it is part of the user-mode is because it contains information that applications access to.  
PEB contains the pointers to the process' DLL list, command line arguments, environment variables, heaps, current working directory, and standard handles.

The second term that it is important to under is "Virtual Address Descriptor" (VAD).  
The memory manager maintains a set of VADs for each process, in order to track, which address space has been reserved in the process address space and which has not.  
VAD is kernel-mode structure.

**Detection of Process Hollowing**In order to detect Process Hollowing on a computer, the most common way is by using Volatility plugins.  
When a malware performed Process Hollowing and Hollows the suspended process Windows removes all the connection between the PE file and the hollowed process, but only from the EPROCESS kernel object and not in the PEB.  
The comparison between the output about all the loaded modules inside a process in the user-mode from the PEB and all the loaded modules according to the kernel-mode objects, will be good indication if Process Hollowing was performed on this process.  
The commands that need to be used in order to do this comparison are "dlllist", which that lists all the loaded modules from the PEB, and "ldrmodules", which lists all the loaded modules from EPROCESS kernel objects.  
There is plugin in volatility that perform the comparison between the loaded modules in the kernel-mode and the user-mode called "HollowFind".

One more way to detect Process Hollowing is by using another Volatility plugin called "malfind", which do comparison between information from the PEB to information in the VADs.

**Summary:**Process Hollowing is advance Process Injection sub-technique, which is used by common and known malwares and attacks.  
This technique can be implemented in several ways, and the main way to detect this attack is by memory forensic.  
As it was written above, the detection is being done by comparison between the PEB in the user-mode and kernel-mode objects.  
Because it is much easier to make changes on user-mode objects it is possible to avoid detection while using this technique.  
In our AV-Evasion platform we will use Process hollowing as one of our modules in order to avoid anti-virus engines.

**References:**

* The Art of Memory Forensic by Michael Hale Ligh, Andrew Case, Jamie Lavy and Aaron Walters
* Mastering Malware Analysis by Alexey Kleymenov and Amr Thabet
* Windows Internals 7th edition by Pavel Yosifovich, Alex Ionescu, Mark E. Russinovich and David A. Solomon
* Threadmap Documentation – by KSL group