**Memory Forensics – Task 4**

Julie Haynes

I organized my report based on the type of evidence that each module might provide. I also included a chart I’ve been using to keep track of the PID/PPID relationship. This does not include every module I ran, just the ones I was able to interpret.

**HIDDEN DLL**

**Ldrmodules** showed potentially hidden DLLs based on the InLoad, InInit, and InMemory showing a value of “False”. The potentially hidden DLLs per process include:

* PID 3824 (explorer.exe)
  + Several DLLs with the .MUI. The MUI extension is used for language specific resource files.
  + Msxml3r.dll, msxml6r.dll – parser libraries
  + Imageres.dll - used to store icons – per <https://www.win7dll.info/imageres_dll.html> there should be 218 icons in this file but I did not find a way to check the number of icons or compare the icons in the hidden file.
* PID 4888 (one of the 2 rogue svchost.exe processes which were both spawned by PID 3824 above)
  + Oleaccrc.dll and oleaccrs.dll.mui
* PID 2980 (StikyNot.exe spawned by PID 3824 above)
  + KernelBase.dll.mui and msxml3r.dll
* PID 644 (svchost.exe)
  + Tracert.exe.mui
  + There is a blank in the “Mapped Path” that shows false for all values – I don’t know what this means –

**CONTACT WITH C&C**

**Netscan** showed that these two processes may have been infected by the C&C server at 194.87.109.183:443 (that was found in pcap file and siem logs). The state in netscan shows closed

* Chrome.exe – PID 2576
* Svchost.exe (PID 2612) – which is one of the 2 rogue Svchost.exe processes spawned by PID 4652/xwgrttjl.exe

**ESCALATING PRIVILEGES**

**Getsids** shows the following processes have elevated privileges, including administrator and remote:

* Obommhdf.exe
* Svchost.exe – specifically the rogue processes PID 2612 and 4104
* Chrome.exe – specifically the PID 2576 that also showed up as contacting the C&C at 194.87.109.183.

**PRESENCE OF PACKED EXECUTABLES**

**Hollowfind and Malfind** showed the following processes contained packed executables based on the presence of the “MZ” tag indicating a PE header:

* Svchost (PID 2612)
* Svchost (PID 4104)
* Tracert (PID 3908)
* StikyNot (PID 3388)
* Notepad (PID4888)
* Chrome.exe (PID 2576)
* GoogleCrashHandler (PID 2556)
* Jusched (PID 2736)
* Armsvc (PID 1152)

**INFECTED PROCESS / DLL or EXTRA DLL**

**Dlllist** showed the following:

* GoogleCrashHandler process (PID 2856) displays a string of Asian language characters directly below the command line. In contrast the other GoogleCrashHandler process displays “Note: Use ldr modules for listing DLLs in WOW 64 processes”. And the other processes in dlllist display either “Service Pack 1” or a similar “Note”. GoogleCrashHandler process not available in baseline for comparison.
* Winlogon process difference between baseline and infected: In baseline, the AUTHZ.dll is loaded by services.exe. In infected, the AUTHZ.dll, while loaded by services.exe it is also loaded by winlogon.exe. (AUTHZ.dll is not loaded by winlogon in the baseline)
* One of the 2 notepad processes in infected (PID4888) contains a lot more dlls than the other notepad process (PID2980) – not sure if normal or sign of extra “not-normal” dlls being added.
* CFGMGR32.dll process loading differences between infected and baseline:
  + Infected: CFGMGR32.dll and WLDAP32.dll are loaded by LSASS, wmpnetwk, StikyNot, Notepad, Chrom, and helpane.
  + Baseline: CFMGR32.dll loaded by svchost, spoolsv, vmtoolsd, wmiprvse, explorer, SearchIndexer, and vssvc

**PERSISTENCE / COPYING DATA (maybe normal, maybe not normal)**

**Filescan** showed:

* The use of “VolumeShadowCopy” (679 times) to copy a lot of information including tasks, files, directories, bitmap, metadata, etc. on HardDiskVolume2. Maybe this was a system back-up but I did not see VolumeShadowCopy used at all on the baseline file.
* Process obommhdf with RWD permissions in the Startup directory – could this be establishing persistence?
* The xwgrttjl.exe file also shows RWD permissions.

**FILE/PROCESS EXECUTION**

**Shimcachemem** showed that the following processes/files executed based on the “True” flag:

* Bilo400
  + Showed “True” for C:\Users\Daniel\AppData\Local\Temp\bilo400.exe
  + Showed “False” for UNC\tsclient\malware-analysis\rig-EK\20a7-12-28-Seamless-campaign-Rig-EK-malware-and-artifacts\bilo400.exe
* Obommhdf (including in Startup folder)
* Xwgrttjl
* Svchost (not confirmed as rogue versions)
* GoogleCrashHandler
* Tracert
* Notepad
* StikyNot

**ROOT KIT**

**Apihooks** showed the following hook types – although I don’t know enough about interpreting the instructions to identify normal vs. evil hooking yet:

* Inline/trampoline hook types. These are available for windows patching but can easy be used by malware by modifying the instructions.
* IAT hook types which can also be leveraged by malware so this is another place I would look at more closely when I can interpret the instructions.

**PID/PPID HIERARCHY CHART**

