Posted on **2013-12-08**

← Previous Next →

Beyond good ol' Run key, Part 5

Time for the 5th part. Today it's about the Phantom DLLs Hijacking (do not confuse it with 'DLL Search Order Hijacking' where order in which paths are searched for is abused).

Windows is a huge operating system and not all components are cleaned up every new release. Thanks to that, there is a lot of legacy code that due to its age and changes in the structure of the operating system no longer work, or are simply left alone – probably just an implementation of the good ol' rule at work – if it doesn't break, don't touch it.

Many Windows applications residing in the system32 directory are not used on daily basis – some of them indeed look like legacy apps (you can still run syncapp.exe on your Windows XP copy to enjoy the Briefcase experience :)), and some of them may be used only in specific configurations (e.g. NT Backup). The mechanisms that some of these legitimate applications use often cater for different scenarios and environments and… thanks to that, they can be often exploited.

Toying around with system32 executables I came across a few legitimate applications that may work as a hidden persistence mechanism . The 'features' I am describing are not new, and are identical with the fxsst.dll trick mentioned in the 3rd part of the series and described by Mandiant in detail. The idea is simple – some programs attempt to load specifically named libraries that are some sort of extension, or plugins and in some cases debugging tools – and they can continue to run, even if these DLLs implementing this extra functionality are not present.

Here's a list of pairs I came across (and I bet there is more; note, some of these may work on other OS versions as well – usually the same code is used across many consecutive OS versions):

ntbackup.exe and edbbcli.dll (Windows XP)
Ntbackup is a Backup Utility for Windows (available on XP, but not 7). When launched it starts a Backup or Restore Wizard.

The interesting thing is that when loaded, this .exe is also attempting to load a library called:

- `%windir%\system32\edbbcli.dll`

Thus, dropping such named DLL and making ntbackup run every time Windows starts either by using Scheduled job or using any of the well-known Autorun mechanisms one can achieve quite a stealthy code execution.

ntbackup.exe and esebcli2.dll (Windows XP)
The situation is exactly the same here; anytime ntbackup launches, it attempts to load the esebcli2.dll DLL from the following location:

- `%windir%\system32\esebcli2.dll`

The legitimate mechanism of loading these two NT Backup DLLs is explained here.

mrt.exe and bcrypt.dll (Windows XP)
MRT is a Malicious Software Removal Tool. When launched, it attempts to load bcrypt.dll; thus, dropping it in the system32 directory on XP will lead to (somehow ironic) execution of the code:

- `%windir%\system32\bcrypt.dll`

The bcrypt.dll doesn't exist on XP, but it does on Windows 7, so this trick will only work on older Windows.

sessmgr.exe and SalemHook.dll (Windows XP)

Yet another binary that can load a targeted DLL is a Remote Desktop Help Session Manager executable on Windows XP. Once launched, it tries to load the following DLL:

- `%windir%\system32\SalemHook.dll`

certreq.exe and msfte.dll (Windows 7)

Certreq is a tool that allows to work with certificates. When launched it attempts to load the following DLL:

- `%windir%\system32\msfte.dll`

certreq.exe and msTracer.dll (Windows 7)

Certreq seems to like non-existing DLLs as it also attempts to load the following DLL:

- `%windir%\system32\msTracer.dll`

FXSCOVER.exe and TPPrnUIENU.dll (Windows 7)

This tool is a Fax Cover Page Editor. When loaded, it attempts to load the following DLL

- `%windir%\system32\spool\DRIVERS\W32X86\3\TPPrnUIENU.dll`

The name of the DLL (TPPrnUIENU.dll) suggests that:

- It is a companion to %windir%\system32\spool\drivers\w32x86\3\TPPrnUI.DLL (there are plenty of %windir%\system32\spool\drivers\w32x86\3\TPPrnUI*.dll files, but not the TPPrnUIENU.dll)
- The ENU in the file name indicates it is a Satellite DLL used in localization and it's providing data for the English language (the other TPPrnUI*.dll present on the system use other language identifiers e.g. TPPrnUIfra.dll, TPPrnUIita.dll)

In theory, these Satellite DLLs are resource-only DLLs, but the TPPrnUI.DLL is using a LoadLibraryW to load them – a big mistake, since the LoadLibraryW calls DllMain of the loaded DLL (LoadLibraryEx with LOAD_LIBRARY_AS_DATAFILE is a better choice here). Hence, a code execution.

dxdiag.exe and DXGIDebug.dll (Windows 7)

dxdiag.exe is a DirectX Diagnostic tool. When executed, it attempts to load the following DLL (for debugging purposes as a part of DXGI framework):

- `%windir%\system32\DXGIDebug.dll`

msinfo32.exe and fveapi.dll (Windows 8.1)

The msinfo32.exe is a system information tool. When executed, it will try to load the following DLL:

- `%windir%\system32\fveapi.dll`

Interestingly, this DLL is present on Windows 7 and Windows 8, but not present on Windows 8.1 (in default installation). The DLLs description is Microsoft Vista BitLocker Drive Encryption API and it offers access to BitLocker API.

narrator.exe and MSTTSLocEnUS.DLL (Windows 8)

Last one on the list is Narrator.exe that attempts to load the following DLL:

- `%windir%\system32\speech\engines\tts\MSTTSLocEnUS.DLL`

Again, then file naming suggests localization gone wrong.

32-bit processes, Wow64.dll and Wow64Log.dll (Windows 7 64-bit)

This is not a process-specific autorun mechanism, but a system-wide way of introducing a DLL into all 32-bit processes running under control of WOW64 by dropping the following 64-bit DLL on the 64-bit Windows 7.

- `%windir%\system32\Wow64Log.dll`

This trick has been discovered and described in detail by Wallied Assar on his blog earlier this year.

While Wow64Log.dll appears to be made available with a purpose of being a legitimate logging feature, it can be abused as a persistent mechanism as well.

This entry was posted in **Anti-Forensics**, **Autostart (Persistence)**, **Compromise Detection**, **Forensic Analysis**, **Malware Analysis** by **adam**. Bookmark the **permalink**.