Malware and the Windows API

Windows API Conventions

Windows API Hungarian Notation

- WORD (w) - 16-bit unsigned value (wVal)
- DWORD (dw) - Double-WORD, 32-bit unsigned value (dwVal)
- Handle (H) – Reference to an object (Hmodule)
- Long Pointer (LP) – Pointer to another type (LPByte)

Windows API Function Suffixes

- A – ANSI strings for params / return values (*CopyFileA*)
  - ANSI – 8-bit characters
- W – WIDE strings for params / return values (*ShellExecuteW*)
  - WIDE – 16-bit characters
- Ex – Extended, has added functionality over normal version of function (*RegSetValueExA*)

Common Windows API Combinations in Malware

Guessing Behavior from API Functions

- Investigating functions in the IAT can imply malware behavior
- Can be even more confident about likely behavior if certain Windows API calls occur sequentially in disassembly

Runtime Linking

- *LoadLibrary* - Load a DLL into a process's memory
- *GetProcAddress* – Gets the address of a function from a DLL in memory
- In combination, can get the address of any function in any DLL on the system
  - Don't need to list desired functions in the IAT

Privilege Escalation

- *OpenProcessToken* – Opens a process's access token (which describes its security context)
- *LookupPrivilegeValue* – Retrieves a locally unique identifier (LUID), which is a struct that represents a specific privilege
- *AdjustTokenPrivileges* – Modifies privileges of an access token
- Usually getting SeDebugPrivilege, which is pretty much admin

Anti-Debugging Timing Checks

- *QueryPerformanceCounter* – Called twice, difference between processor's performance counter at each call is calculated

- *GetTickCount* – Called twice, difference between number of milliseconds since computer boot is calculated

Other Anti-Debugging API Functions

- *IsDebuggerPresent* – Checks the current process's Process Environment Block (PEB) for the status of IsDebugged field

- *CheckRemoteDebuggerPresent* – Checks the PEB of any process on the machine for the status of the IsDebugged field

Even More Anti-Debugging API Functions

- *NtQueryInformationProcess* – Gets information about a process given its handle. When passed the ProcessDebugPort parameter, returns the debug status.

- *SetLastError*, *OutputDebugString, GetLastError* – Sends a string for a debugger to display. If no debugger is present, the current error code has changed.

Process Injection

- *VirtualAlloc* – Allocate space in an external process's memory
- *WriteProcessMemory* – Write data (executable code to be executed as a thread) to the allocated space
- *CreateRemoteThread* – Execute the injected code as a thread belonging to the victim process

Download + Execute

- *URLDownloadToFile* – Download a file from the internet and save it to disk

- *WinExec* / *ShellExecute* – Execute the downloaded file

Polling Keylogger

- *FindWindow* + *ShowWindow* / *GetForegroundWindow* – Gets a handle to a specific window / the window in the foreground
- *GetKeyState* / *GetAsyncKeyState* – Gets whether a key is being pressed
- Usually found in a nested loop. The outer loop gets a window and the inner polls the state of each key

Hooking Keylogger

- *SetWindowsHook* – Creates a Windows hook that gets notified when a keyboard event happens.

- *GetMessage* – Called in a loop to retrieve keyboard event messages

Taking Screenshots

- *GetDesktopWindow* – Get a handle to the desktop window, which contains the entire screen
- *BitBlt, GetDIBits* – Given a handle to a window, copy pixels to a destination buffer
- Often seen with other functions, such as CreateFile (to save the screenshot)