Can all commands be watched with WinDbg

Posted by Will Steele - 29 Dec 2009 - 23:16

This is a very basic question. I have not been able to figure out the answer and I have been curious about this for a while. If I open an executable from WinDbg, is there a way to watch every single action performed by the program as it launches? If so, is it only in assembly, or, is there a way to approach the call by call from say a module function call level? To me, this would be a great way to figure out what applications are really doing when they start, but, I am not sure if this is even really something valid to think about with WinDbg.

==========

Re:Can all commands be watched with WinDbg

Posted by Robert Kuster - 30 Dec 2009 - 00:35

Hi Will,

yes it can be done with WinDbg. Let's start with the File->Open Executable from WinDbg's menu; this way WinDbg starts your application and actually stops after several system DLLs (for example kernel32, advapi, gdi32) have been loaded into the process. Unfortunately this is not early enough for what you intend to do, so let us first setup our environment:

1) Open GFlags.exe (installed alongside WinDbg.exe into the same directory).
> Image File (tab)
- > Image -> enter the patch of your executable, i.e.: "C:CrashMe.exe"
- > Debugger -> enter the path to WinDbg, for example: "C:Program FilesDebugging Tools for Windows (x86)windbg.exe"

2) Start CrashMe.exe
Note that WinDbg is attached right away. But as the attach still happens (too) late during our process-start, we have to change one more thing.

> WinDbg (Menu) -> Debug -> Event Filters -> select the "Load module" event AND change Execution to "Enabled"
> Close the dialog and exit WinDbg (don't forget to save the workspace).

3) Start CrashMe again
This time WinDbg is attached right after ntdll.dll has been loaded into our newly created address space. Now you can debug the windows loader which is actually implemented in large part in ntdll.dll.

Note that ntdll:
a) is involved in creating a virtual address spaces for your process
b) loads kernel32, gdi32 and other system DLL into the just created virtual address space
c) creates the main/first thread with a BaseAddress in kernel32
d) finally the main thread executes a jump from kernel32 to the entry function of your application (main, WinMain, etc)

If I open an executable from WinDbg, is there a way to watch every single action performed by the

program as it launches? Yes, you will see everything though mainly a lot of disassembly.

If so, is it only in assembly, or, is there a way to approach the call by call from say a module function call level?Well, yes and no. It depends on what you mean with function call level.

While you will be able to see many internal MS function names in ntdll, kernel32, and other system DLLs, you wont see function parameters or local variables for these functions.
Note that MS releases only public symbols for their modules. For local variables or function parameters one would need private PDB symbols. Again welcome to read more on symbols at WinDbg. From A to Z!, slides 11-14.

I hope this helps.

Warm Regards, RK

PS: Oh, there is one more candy I forgot about. In GFlags you could also enable "Show loader snaps". With this flag enabled ntdll would emit additional loader notifications to WinDbg.

==========

Re:Can all commands be watched with WinDbg

Posted by Will Steele - 30 Dec 2009 - 02:34

Thanks Robert. You've given me a lot to work with. I am still in the "wide-eyed" stage of figuring out very basics. These things help me get a grasp much more quickly.

==========