

Forum:[Hack Forums](#)

Thread: [API hooking bypass: What about WoW64?](#)

wayofAK - May 29, 2020, 1:35 pm

Hello, I have a question regarding NT API un-hooking. There are a lot of ways explained in various write-ups, like this one [0]

As far as I got, one way is to get into the syscalls, on our own, without depending on ntdll. This way, we're crossing to kernel mode without any help of the ntdll which is (hooked) by an AV, like explained here [1].

This works well when talking about a 32 bit process on 32 windows or 64 bit process on 64 windows. There, u can get the syscall, put it urself in ur code, and bypass the hook. As we are directly using syscalls in the loaded ntdll.

For 32 process in 64 windows. This isn't made directly. ntdll (32 bit) points to WoW64 transition in another dll which points after some jumps to the one which calls syscalls (ntdll 64 bit as we are on 64 windows) see [2]

Knowing this I have some questions I hope any one with enough exp. to answer.

- 1- Where does the AV hooks in this case? the 64 bit ntdll (at /system32) or 32 bit one (at /WoW64) which will finally land to the 64 one. (I guess it will be at the 64 one, but not sure)
- 2- Does this mean, I can drop the dependency on intermediate WoW64 dlls and just call sys calls according to the windows 32/64 myself ? (so the plan is to check windows arch, if 64 use syscall with 64 codes else use syscall with 32 codes)
- 3- What are AVs which uses this technique to test my exes on?

[0] <https://hackforums.net/showthread.php?tid=5655530&pid=55198692#pid55198692>

[1]

<https://outflank.nl/blog/2019/06/19/red-team-tactics-combining-direct-system-calls-and-srdr-to-bypass-av-edr/>

[2] <https://medium.com/@fsx30/hooks-heavens-gate-a-wow64-hooking-technique-5235e1aeed73>

akali - May 29, 2020, 2:55 pm

A/V would hook at the callgate/transition from WOW64 to x64. Since the ordinal is passed along with the arguments into x64.

Although I doubt many Av even bother with usemode and hook directly in the kernel probably the SSDT. I can't remember if this sets off patchguard or not.

If you don't want to deal with a dropper and carrying extra binaries in your resources..then only produce 64 bit? I don't know why WOW64 is even supported anymore, I highly doubt you can find a windows machine running a 32bit processor... Still if you want to take that route then you need a 32 bit module that can make x64 calls in the case you're trying to load memory into a 64 bit module.

Don't know. A/V probably aren't that good to begin with. They mostly just check signatures and hashes.

Windows should just prevent I/O access to any directory unless explicitly allowed by the user when the binary is executed.. This is what MacOS does.

