GitHub - hfiref0x/TDL: Driver loader for bypassing Windows x64 Driver Signature Enforcement

By hfiref0x

Archived: 2026-04-05 15:50:40 UTC

Driver loader for bypassing Windows x64 Driver Signature Enforcement

For more info see

- Defeating x64 Driver Signature Enforcement http://www.kernelmode.info/forum/viewtopic.php?f=11&t=3322
- WinNT/Turla http://www.kernelmode.info/forum/viewtopic.php?f=16&t=3193

System Requirements and limitations

- x64 Windows 7/8/8.1/10.
- TDL designed only for x64 Windows, Vista not listed as supported because it is obsolete.
- Administrative privilege is required.
- Loaded drivers MUST BE specially designed to run as "driverless".
- No SEH support for target drivers.
- No driver unloading.
- Only ntoskrnl import resolved, everything else is up to you.
- Dummy driver examples provided.

You use it at your own risk. Some lazy AV may flag this loader as malware.

Differences between DSEFix and TDL

While both DSEFix and TDL uses advantage of driver exploit they completely different on way of it use.

- DSEFix manipulate kernel variable called g_CiEnabled (Vista/7, ntoskrnl.exe) and/or g_CiOptions (8+. CI.DLL). Main advantage of DSEFix is it simplicity - you turn DSE off - load your driver (or patched one) and nothing else required. Main disadvantage of DSEFix is that on the modern version of Windows (8+) g_CiOptions variable is subject of PatchGuard (KPP) protection, which mean DSEFix is a potential BSOD-generator.
- TDL does not patch any kernel variables, which makes it friendly to PatchGuard. It uses small shellcode which maps your driver to kernel mode without involving Windows loader (and as result without triggering any parts of DSE) and executes it. This is main advantage of TDL - non invasive bypass of DSE. There are many disadvantages however - the first and main -> your driver MUST BE specially created to run as "driverless" which mean you will be unable to load *any* driver but only specially designed. Your driver will exist in kernel mode as executable code buffer, it won't be linked to PsLoadedModuleList, there will be

other limitations. However this code will work at kernel mode and user mode application will be able communicate with it. You can load multiple drivers, of course if they are not conflict with each other.

How it work

It uses WinNT/Turla VirtualBox kernel mode exploit technique to write code to the kernel memory and after execute this code. TDL uses custom bootstrap shellcode to map your specially designed driver and call it entry point (DriverEntry), note that DriverEntry parameters will be invalid and must not be used. Examples of specially designed drivers available as DummyDrv and DummyDrv2. Your DriverEntry will run at IRQL PASSIVE_LEVEL up to Windows 10 RS1. Starting from Windows 10 RS2 your DriverEntry code runs on IRQL DISPATCH_LEVEL.

Build

TDL comes with full source code. In order to build from source you need Microsoft Visual Studio 2015 U1 and later versions. For driver builds you need Microsoft Windows Driver Kit 8.1 and/or above.

Instructions

- Select Platform ToolSet first for project in solution you want to build (Project->Properties->General):
  - v120 for Visual Studio 2013;
  - v140 for Visual Studio 2015;
  - v141 for Visual Studio 2017.
- For v140 and above set Target Platform Version (Project->Properties->General):
  - If v140 then select 8.1 (Note that Windows 8.1 SDK must be installed);
  - If v141 then select 10.0.17763.0 (Note that Windows 10.0.17763 SDK must be installed).

Remove linker option /NOCOFFGRPINFO where it unsupported/unavailable.

Deprecation

TDL based on old Oracle VirtualBox driver which was created in 2008. This driver wasn't designed to be compatible with newest Windows operation system versions and may work incorrectly. Because TDL entirely based on this exact VirtualBox driver version LPE it is not wise to use it on newest version of Windows. Consider this repository as depricated/abandonware. The only possible updates can be related only to TDL loader itself.

Authors

Credits

- R136a1
- N. Rin

Source: https://github.com/hfiref0x/TDL