Windows kernel rootkits techniques and analysis

Table of Contents

Instructor

Bruce Dang

Duration

4 days

Class description

This class is tailored for malware analysts, system developers, forensic analysts, incident responders, or enthusiasts who want to analyze Windows kernel rootkits or develop software for similar tasks. It introduces the Windows architecture and how various kernel components work together at the lowest level. It discusses how rootkits leverage these kernel components to facilitate nefarious activities such as hiding processes, files, network connections, and other common objects. As part of the analytical process, we will delve into the kernel programming environment; we will implement some kernel-mode utilities to aid our understanding.

Needless to say, the class will contain many hands-on labs and exercises using real-world rootkits. There are no made-up examples in the class.

What you should expect

After this class, you should have a systematic understanding of Windows kernel to analyze rootkits and develop kernel-mode utilities (or even products!) for your job. In addition, you will be able read and understand research on Windows kernel and related subjects. You will no longer feel intimiated by the kernel.

In previous classes, practically all students were able to analyze kernel rootkits and develop drivers on their own at the end of the course. Many of these students have never written a driver before in their life and they felt comfortable doing it after the third day. Here are some examples of what some students accomplished after class: analyzed well-known kernel APTs, analyzed Windows PatchGuard, developed a driver to remap keys, researched into hypervisor development.

Training topics

- x86/x64 architecture and system facilities
- Windows kernel architecture
- Debugging facilities
- Data structures
- Memory management
- Process and threads
- Files and networking
- User-kernel facilities
- Drivers
- Kernel/exploit development

Intended audience

Malware analysts, systems programmer, forensic analysts, security engineers, network security analysts, kernel enthusiasts.

Prerequisites

In order to get the most out of this class, you need to have some programming experience; if you are not comfortable with that, you can still understand the material and immediately apply it to your daily job, however you might need to work extra hard in class.

Hardware and software requirement

Hardware

- Laptop running Windows as the host OS

You will spend a lot of time tracing, debugging, and developing rootkits/drivers. Hence, for the purpose of this class, please bring a laptop running Windows as the host OS. In previous courses, some students brought MacBooks (running OS X and VMWare Fusion) and end up spending a lot of time fighting with configuration settings; while it is technically possible to have two Windows VMs (one target, one host), it is painfully slow and unnatural.

The class involves toggling between various versions of Windows and snapshotting/restoring VMs. Hence, if your laptop has an SSD, you will have a better experience.

Software

You should have the following software installed on your host OS:

- Visual Studio 2015. It does not matter what edition you use. The Community Edition is free and we will use that in the class.
- Windows WDK 10. https://msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx
- Windows SDK 10. https://dev.windows.com/en-us/downloads/windows-10-sdk
- NASM. http://www.nasm.us
- IDA Pro (decompilers are not required, but x64 support would be good)
- VMWare Workstation. You can just use the free 30-day trial version

Do not make the mistake of installing these software while in class as it will take many hours.

In addition, you will need to have VM images for these OSes:

- Windows 7 x86/x64 (you will need both because we will deal with both 32 and 64bit rootkits/drivers)

About the instructor

Bruce Dang is an information security researcher with interests in low-level systems. He is currently working at Veramine trying to make the world a safer place. He previously worked as a senior security development engineer lead at Microsoft; his team's focus spans all things product-security related from hardware, OS, and web services. He specializes in reverse engineering and Windows kernel-level security projects. Before Microsoft, he worked as a developer in the financial sector. He was the first person to publicly discuss techniques of analyzing file format based exploits and has patents in the area of generic shell code and exploit detection. His public research includes Office exploit analysis, ROP detection, shell code detection, and kernel driver decompilation techniques; on the malware side, he is known for first analyzing vulnerabilities in the Stuxnet worm. He has spoken at major security conferences worldwide, i.e., REcon (Canada), Blackhat (Vegas and Tokyo), Chaos Computer Club (Germany), Computer Antivirus Research Organization (Hungary), etc. In addition to sharing his knowledge at public conferences, he has also provided private training and lectures to government agencies. He is also the author of the best-selling reverse engineering textbook, Practical Reverse Engineering: x86, x64, Windows kernel, and obfuscation, published by John Wiley & Sons.

Last updated 2017-03-02 18:54:52 PST