Windows® Internals for Forensic Analysts

Overview

The Windows operating system is a large place and the amount of information available from a system can be daunting. In order to understand where to look for interesting information and how to interpret it, one needs a solid understanding of the base architectural concepts that underlie the system as well as the data structures that "glue" it all together.

During this intense, four day seminar, the student will have a chance to explore topics such as virtual memory, the impact of running Windows under a hypervisor, the on disk structure of the most common Windows file systems, and the registry through a mix of concentrated lectures and hands-on lab experiments.

Seminar Formats

This seminar is available as a 4 day lecture with lab.

Target Audience

This class is directed at anyone that wants to gain a deeper understanding of the Windows operating system, but particularly those looking to use this information for the purposes of forensic analysis and incident response. This course is designed to help the student intelligently use, but not rely on, their existing set of analysis utilities.

Prerequisites

This is not a seminar for beginners without knowledge or experience in either operating systems or firmware. Rather, it is an intense, architectural study of specific parts of the Windows O/S, interspersed with practical lab exercises. Students attending this seminar will need a good understanding of general O/S concepts. They also will be well served by having basic knowledge of Windows O/S architectural concepts (such as can be gained by reading Windows Internals (by Russinovich, Solomon, and Ionescu). Because it is a hands-on seminar, students need to be familiar with scripting languages such as Python, as well as the basic ability to use a Windows system.

Hardware Requirement: Attendance at this seminar requires each participant to bring their own laptop in a configuration capable of running a virtual machine.

Copyright (c) 2017 OSR Open Systems Resources, Inc.

Seminar Outline

1. **Windows OS Architecture Overview**
   An overview of the general architecture of the Windows operating system.

   Lab: No Lab

2. **Physical Memory Acquisition Techniques**
   A description of the various techniques used to acquire physical memory images of systems. Raw memory dumps, crash dump files, hibernate files, and paging files. Common acquisition tools discussed.

   Lab: Utilize the tools discussed to acquire and explore physical memory images from the lab machine.

3. **Introduction to WinDbg**
   WinDBG is THE tool for analyzing physical memory dumps adhering to the Windows crash dump file format. Installation, configuration, and use of WinDBG for analysis of crash dump files are covered. Also includes a discussion of the WinDBG scripting language and COM based extension interface.

   Lab: Configure WinDBG for crash dump analysis and explore an image acquired by one of the previously discussed tools.

4. **Windows Key Data Structures**
   In this module, we discuss the fundamental Kernel Mode data structures in Windows, such as the KPCR, KTHREAD, EPROCESS, etc. This includes coverage of key Dispatcher, Control, and Executive object types.

   Lab: Using WinDBG, explore the data structures discussed in this section.

5. **Virtual and Physical Memory Concepts**
   How and why Windows implements virtual memory for user applications and the operating system. Page Tables, Page Directories, the PFN, and the VAD. Memory manager policy, including working sets.

   Lab: Perform a manual virtual to physical translation in PAE and non-PAE modes. Explore physical memory and the page frame database.

6. **O/S Entry: Traps, Interrupts and System Services**
   The main methods of O/S entry are discussed, including traps, interrupts, and system services. The I/O subsystem is discussed, including the role of Kernel Mode drivers.
   Lab: View the contents of the SSDT on the x86 and x64. Provide a WinDBG script to retrieve the symbolic names of the entries of the SSDT on the x64. Explore the I/O subsystem.

Copyright (c) 2017 OSR Open Systems Resources, Inc.

7. **BIOS and System Startup**
   Describe how modern BIOS's affect the execution of the Windows O/S. ACPI and the ACPI Source Language is discussed. System Management Mode (SMM) is discussed. Windows' interaction with the BIOS during the boot process is discussed.

   Lab: Explore the ACPI BIOS on the lab machine.

8. **Windows Security**
   The Windows security subsystem is discussed. ACLs, tokens, privileges, and components of the TCB.

   Lab: Explore security descriptors and tokens. View the impact of User Access Control on user tokens.

9. **Process and Thread Creation**
   How processes and threads are created and destroyed in Windows. Thread scheduling is discussed.

   Lab: View and modify the behavior of existing threads and processes on a live system.

10. **Portable Executable (PE) File Format**
    All executable images in Windows including applications, device drivers, and DLLs adhere to the Portable Executable (PE) File Format. Import tables, export tables, and the on disk and in memory formats are discussed.

    Lab: View the PE file format. Locate the functions imported by a kernel mode driver running on the target machine.

11. **Physical Media File System Basics**
    Brief introduction to basic file system concepts, including metadata and space allocation. Differences between local and network file systems is discussed.

    Lab: No Lab

12. **File Allocation Table (FAT) File System Internals**
    A description of the FAT file system and its associated metadata. FAT16, FAT32, and ExFAT are covered.

    Lab: Recover the contents of a deleted file from a FAT32 volume.

13. **NTFS Internals**
    A description of the NTFS file system and its associated metadata.

    Lab: Recover the contents of a deleted file from an NTFS volume.

Copyright (c) 2017 OSR Open Systems Resources, Inc.

14. **Windows Registry Internals**
    The Windows registry is a rich database of potentially interesting information to an analyst. This module contains a description of both the in memory and on disk structure of the Windows registry.

    Lab: Recover the contents of a registry key without the use of the Registry Editor.

15. **Virtualization**
    As virtualization continues to increase in popularity, an understanding of virtualization techniques and hypervisors becomes important. Virtualization techniques are discussed in general as well as specific Windows implementation details. Enlightenments are discussed.

    Lab: Investigate a Windows installation running under a hypervisor.

16. **Stealth Techniques and Counter Attacks**
    Based on the foundations covered within the course, a discussion of the currently in use stealth techniques are discussed. Included are methods used for interception of API and I/O operations, process hiding, file hiding, and persisting across reboots. Features added to Windows to counter these attacks are discussed, including no-execute, PatchGuard, and Address Space Layout Randomization (ASLR).

    Lab: Explore the topics discussed in this section. Detect the presence of code on the machine employing one or more of these techniques.

17. **Kernel Mode Software Implementation Basics**
    As a basis for more in depth analysis of Kernel Mode components, the basics of Kernel Mode software on Windows is discussed. The common entry points of drivers are discussed, including DriverEntry, AddDevice, and the Dispatch Entry points. Common data structures used by drivers and programming patterns are discussed, as well as code signing requirements on 32bit and 64bit versions of Windows.

    Lab: Become familiar with the structure of a Kernel Mode driver by live debugging driver at the source level.

Copyright (c) 2017 OSR Open Systems Resources, Inc.