Windows Internals

Course Summary Table

Duration:	5 Day
Target Audience:	Windows developers and/or researches, interested in writing better programs, by getting a deeper understanding of the internal mechanisms of the windows operating system.
Objectives:	Understand the underlying mechanism and advanced services of the windows OS and use that knowledge to write better and more efficient programs on windows 7 and later
Pre Requisites:	Basic knowledge of OS concepts and architecture Recommended: practical experience developing windows application C/C++ knowledge is an advantage

Instructor: **Pavel Yosifovich**

Abstract

The Windows OS exposes many advanced services to system programmers through the Windows API, and to device driver writers through the Kernel API. The .NET framework wraps these services and runs on top of the Windows API and the Kernel.

Good knowledge of what's going on under the hood of the OS, which services are available and how to best utilize them helps in building better and more efficient software for Windows. Those working in the Cyber security space can greatly benefit from the course as it looks at all major Windows mechanisms. Lab exercises are used to reinforce the theoretical material.

Syllabus

- Module 1: System Architecture
  - Brief Windows NT History
  - Basic Concepts
  - Windows Versions
  - Tools
  - Processes and Threads
  - Virtual Memory
  - User mode vs. Kernel mode
  - Objects and Handles
  - Architecture Overview
  - Key Components
  - APIs: Win32, Native, .NET, COM, WinRT
  - User/kernel transitions
  - Introduction to WinDbg
  - System Processes

- Lab: Task manager, Process Explorer, WinDbg

- Module 2: Processes & Jobs
  - Processes Overview
  - Process Internals & Data Structures
  - Creating and terminating processes
  - The loader
  - DLL explicit and implicit linking
  - Process attributes
  - Protected processes and PPL
  - UWP Processes
  - Minimal and Pico processes
  - Jobs
  - Nested jobs
  - Introduction to Silos

- Module 3: Threads
  - Thread basics
  - Creating threads
  - Processor Groups
  - Thread Priorities
  - Thread Scheduling
  - Thread Stacks
  - Thread States
  - Affinity
  - CPU Sets
  - Direct switch
  - Thread Synchronization

- Module 4: Kernel Mechanisms
  - Trap Dispatching
  - Interrupts
  - Interrupt Request Levels (IRQLs)
  - Deferred Procedure Calls (DPCs)
  - Exceptions
  - System Crash
  - Object Management
  - Objects and Handles
  - Sharing Objects
  - Synchronization
  - Synchronization Primitives
  - Signaled vs. Non-Signaled
  - High IRQL Synchronization
  - Windows Global Flags
  - Kernel Event Tracing
  - Wow64

- Module 5: Memory Management
  - Overview

  - Small, large and huge pages
  - VMM Services
  - Page states
  - Address Space Layout
  - Address Translation Mechanisms
  - The Heap Manager
  - APIs in User mode and Kernel mode
  - Page Faults
  - Page Files
  - Workings Sets
  - Memory Mapped Files
  - Page Frame Database
  - Other memory management features (ASLR, compression, enclaves)

- Module 6: I/O System
  - I/O System overview
  - I/O Function
  - Device Drivers
  - Plug & Play
  - The Windows Driver Model (WDM)
  - The Kernel Mode Driver Framework (KMDF)
  - I/O Processing and Data Flow
  - IRPs
  - Power Management
  - Driver Verifier
  - Writing a Software Driver (if time permits)

- Module 7: Security
  - Security components
  - Virtualization Based Security
  - Credential guard
  - User Access Control (UAC)
  - Integrity Levels
  - Protecting objects
  - SIDs
  - Tokens
  - Privileges
  - ACLs
  - Access checking
  - AppContainers
  - Process mitigations