Windows Internals Training

Instructor: Pavel Yosifovich

- Public 5-day remote class
- Dates: June 11, 12, 14, 19, 20
- Time: 8 hours / day. Exact hours TBD
- Price: 1950 USD
- Register by emailing zodiacon@live.com and specifying "Windows Internals Training" in the title
    - Provide names of participants (discount available for multiple participants from the same company), company name and time zone.
    - You'll receive instructions for payment and other details
- Virtual space is limited!

Objectives:	Understand the Windows system architecture Explore the internal workings of process, threads, jobs, virtual memory, the I/O system and other mechanisms fundamental to the way Windows works Write a simple software device driver to access/modify information not available from user mode
Target Audience:	Experienced windows programmers in user mode or kernel mode, interested in writing better programs, by getting a deeper understanding of the internal mechanisms of the windows operating system. Security researchers interested in gaining a deeper understanding of Windows mechanisms (security or otherwise), allowing for more productive research
Pre-Requisites:	Basic knowledge of OS concepts and architecture. Power user level working with Windows Practical experience developing windows applications is an advantage C/C++ knowledge is an advantage

- **Module 1: System Architecture**
    - Brief Windows NT History
    - Windows Versions
    - Windows 10 and Future versions
    - Tools: Windows, Sysinternals, Debugging Tools for Windows

- Processes and Threads
- Virtual Memory
- User mode vs. Kernel mode
- Objects and Handles
- Architecture Overview
- Key Components
- User/kernel transitions
- APIs: Win32, Native, .NET, COM, WinRT
- Introduction to WinDbg
- Lab: Task manager, Process Explorer, WinDbg

- **Module 2: Processes & Jobs**
  - Process basics
  - Creating and terminating processes
  - Process Internals & Data Structures
  - The loader
  - DLL explicit and implicit linking
  - Process and thread attributes
  - Protected processes and PPL
  - UWP Processes
  - Minimal and Pico processes
  - Jobs
  - Nested jobs
  - Introduction to Silos
  - Lab: viewing process and job information; creating processes; setting job limits

- **Module 3: Threads**
  - Thread basics
  - Creating and terminating threads
  - Processor Groups
  - Thread Priorities
  - Thread Scheduling
  - Thread Stacks
  - Thread States
  - CPU Sets
  - Other mechanisms: Autoboost, Direct Switch, Deep freeze
  - Thread Synchronization
  - Lab: creating threads; thread synchronization; viewing thread information; CPU sets

- **Module 4: Kernel Mechanisms**
  - Trap Dispatching
  - Interrupts & Exceptions
  - System Crash
  - Object Management
  - Objects and Handles
  - Sharing Objects
  - Synchronization
  - Synchronization Primitives
  - Signaled vs. Non-Signaled

- Windows Global Flags
- Kernel Event Tracing
- Wow64
- Lab: Viewing Handles, Interrupts; creating maximum handles

- **Module 5: Memory Management**
  - Overview
  - Small, large and huge pages
  - Page states
  - Address Space Layout
  - Address Translation Mechanisms
  - Heaps
  - APIs in User mode and Kernel mode
  - Page Faults
  - Page Files
  - Commit Size and Commit Limit
  - Workings Sets
  - Memory Mapped Files (Sections)
  - Page Frame Database
  - Other memory management features (ASLR, compression, enclaves)
  - Lab: committing & reserving memory; using shared memory; viewing memory related information

- **Module 6: Management Mechanisms**
  - The Registry
  - Services
  - Starting and controlling services
  - Windows Management Instrumentation
  - Lab: Viewing and configuring services; Process Monitor

- **Module 7: I/O System**
  - I/O System overview
  - Device Drivers
  - The Windows Driver Model (WDM)
  - The Windows Driver Foundation (WDF)
  - WDF: KMDF and UMDF
  - I/O Processing and Data Flow
  - IRPs
  - Plug & Play
  - Power Management
  - Driver Verifier
  - Writing a Software Driver
  - Labs: viewing driver and device information; writing a software driver

- **Module 8: Security**
  - Security Components
  - Virtualization Based Security

- Protecting objects
- SIDs
- Tokens
- ACLs
- Privileges
- Access checks
- AppContainers
- Logon
- User Access Control (UAC)
- Process mitigations
- Lab: viewing security information