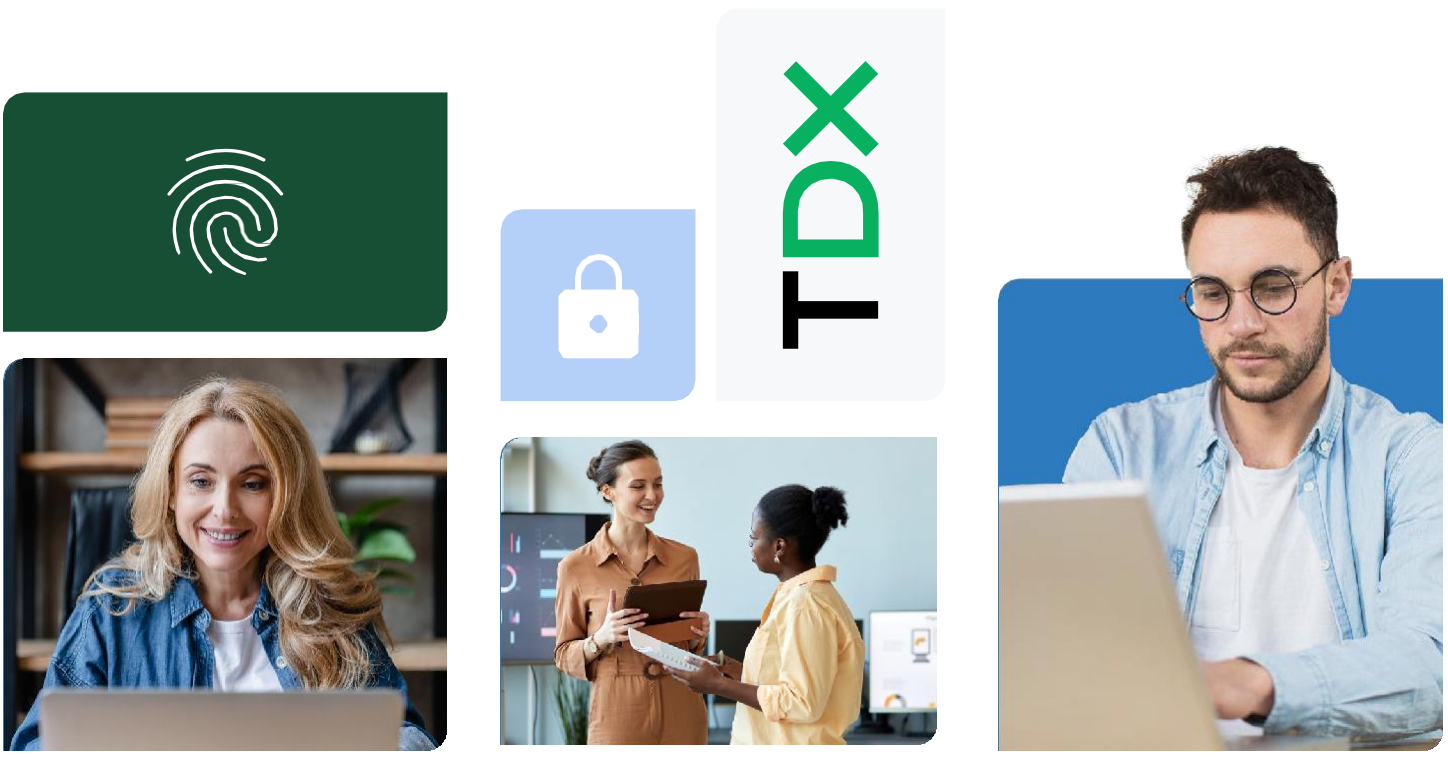
ThriveDX™

Windows Kernel Programming

TDXAISD-104



ThriveDX Windows Kernel Programming

Time Commitment
5 days (total of 40 hours / 8 hours per day)

Skill Level
Professional Level

Course Category
AI | Software Development

The cyber security industry has grown considerably in recent years, with more attacks that are sophisticated and consequently more defenders. To have a fighting chance against sophisticated attacks, kernel mode drivers must be employed, where nothing (at least nothing from user mode) can escape its eyes.

The course provides the foundations for the most common software device drivers that are useful not just in cyber security, but also other scenarios, where monitoring and sometimes prevention of operations is required. Participants will write real device drivers with useful features they can then modify and adapt to their particular needs.

Prerequisites

- At least 2 years of experience working with the Windows API
- Basic understanding of Windows OS concepts such as processes, threads, virtual memory and DLLs

Objectives

- Understand the Windows kernel driver programming model
- Write drivers for monitoring processes, threads, registry and some types of objects
- Use documented kernel hooking mechanisms
- Write basic file system mini-filter drivers

Target Audience

Experienced windows developers, interested in developing kernel mode drivers

Program Structure

- Module 1: Windows Internals quick overview
- Processes and threads
- System architecture
- User / kernel transitions
- Thread synchronization
- Virtual memory
- Objects and handles
- Summary

Module 2: The I/O System

- I/O System overview
- Device Drivers
- The Windows Driver Model (WDM)
- The Kernel Mode Driver Framework (KMDF)
- Other device driver models
- Driver types
- Software drivers
- Driver and device objects

TDX

- I/O Processing and Data Flow
- Accessing devices
- Asynchronous I/O
- Summary

Module 3: Kernel programming basics

- Installing the tools: Visual Studio, SDK, WDK
- C++ in a kernel driver
- Creating a driver project
- Building and deploying
- The kernel API
- Strings
- Linked Lists
- The DriverEntry function
- The Unload routine
- Installation
- Deployment
- Summary
- Lab: create a simple driver; deploy a driver

Module 4: Building a simple driver

- Creating a device object
- Exporting a device name
- Building a driver client
- Driver dispatch routines
- Introduction to I/O Request Packets (IRPs)
- Completing IRPs
- Handling DeviceIoControl calls
- Testing the driver
- Debugging the driver
- Using WinDbg with a virtual machine
- Summary
- Lab: open a process for any access; zero driver; debug a driver

Module 5: Kernel mechanisms

- Interrupt Request Levels (IRQLs)
- Interrupts
- Deferred Procedure Calls (DPCs)
- Asynchronous Procedure Calls (APCs)
- Dispatcher objects
- Low IRQL Synchronization
- Spin locks
- Work items
- Summary

Module 6: Process and thread monitoring

- Motivation
- Process creation/destruction callback
- Specifying process creation status
- Thread creation/destruction callback
- Notifying user mode
- Writing a user mode client
- Preventing potentially malicious processes from executing
- Summary
- Lab: monitoring process/thread activity; prevent specific processes from running

Module 7: Object and registry notifications

- Lab continuation from day 3
- Process/thread object notifications
- Pre and post callbacks
- Registry notifications
- Performance considerations
- Reporting results to user mode
- Summary
- Lab: protect specific process from termination; simple registry monitor

Module 8: File system mini filters

- File system model
- Filters vs. mini filters
- The Filter Manager
- Filter registration
- Pre and Post callbacks
- File name information
- Contexts
- File system operations
- Filter to user mode communication
- Debugging mini-filters
- Summary
- Lab: protect a directory from write; hide a directory; backup file before deletion

"Participants will write real device drivers with useful features they can then modify and adapt to their particular needs"

The ThriveDX Advantage

1 Handcrafted Training Programs

2 State-Of-The-Art Learning Materials

3 Israel's Premier Training Center

4 Fueled by Industry Leading Experts

5 Over 20 Years of Proven IT Education Success