CodeMachine Kernel Security Course Reading List

Reference Books

1. ### Windows Internals Part I and Part II, 6th Edition
   This book applies to Windows 7. Although still very relevant today, quite a few things have changed in more recent versions of Windows. Some of the hands-on labs, especially the ones that involve the kernel debugger, may not work at all or may not display the same results as shown in the book. This is due to changes in the debugger extensions and changes in operating system internal data structures
2. ### Inside Windows Debugging
   The focus of this book is on user mode debugging. There is only a single chapter from this book that applies to kernel debugging, but it is a good book to get introduced to WinDBG.
3. ### Rootkits - Subverting the Windows Kernel
   This book is applies to Windows XP. More recent versions of Windows have built-in mitigations for a lot of the techniques discussed in this book. It is nevertheless useful to understand these techniques.
4. ### The Rootkit Arsenal, 2nd Edition
   This book covers only certain aspects of rootkits but it is the best book on rootkits available today
5. ### Programming the Windows Driver Model, 2nd Edition
   This book is from the Win9x/2000/XP time period. Please DO NOT focus on the code samples, just read and understand the theory.

Kernel Internals

Kernel Debugger

Debugger Package, Debugger Parameters, Kernel Mode Debugging, Debugger Symbols, Debugger Command Types, Debugger Command Reference.

[2] Chapter 2: Getting Started
[4] Chapter 5: Tools of the Trade

Kernel Architecture

NTOSKRNL, HAL & Drivers, Processes and Threads, System & System Idle Process, Process and Thread Data Structures, System Calls, System Service Dispatching , KUSER_SHARED_DATA, User vs Kernel Mode Execution,

[1] Chapter 2: System Architecture
[1] Chapter 3: System Mechanisms (System Service Dispatching)
[1] Chapter 5: Process, Threads, Jobs (Process Internals, Thread Internals)
[4] Chapter 4: System Briefing (4.6)

Execution Contexts

Kernel Processor Control Region (KPCR), Interrupt Request Levels (IRQL), Interrupt Service Routines (ISR), Deferred Procedure Calls (DPC), Asynchronous Procedure Calls (APC), System Worker Threads,

[1] Chapter 3: System Mechanisms (Trap Dispatching, Interrupt Dispatching, System Worker Threads)

Synchronization

Dispatcher Objects, Thread Waits, Interlocked Operations, Mutexes, Fast Mutexes & Guarded Mutexes, Executive Resources, Spin Locks.

[1] Chapter 3: System Mechanisms (Synchronization)

Memory Management

Virtual Address Space, Virtual Address Descriptors, Kernel Virtual Address Space, Address Translation, PTE and Session Space, PFN Database, System Cache, Kernel Mode Stacks, Kernel Pools, Memory Descriptor Lists, Memory Mapping,

[1] Chapter 10: Memory Management
[1] Chapter 11: Cache Manager (Cache VM Management, Cache Data Structures, File System Interfaces)
[4] Chapter 4: System Briefing (4.1 - 4.5)

Objects & Handles

Object Manager, Object Namespace, Objects and Handles, Object Layout, Object Header, Object Types, Object Type Callbacks.

[1] Chapter 3: System Mechanisms (Object Manager)

I/O Management

Driver Architecture, I/O Manager Data Structures, Hardware Device Tree, Driver Types (Bus, Function, Filter), Device Types (FDO, PDO, FiDO), Filter Drivers, Driver Entry Points, Driver Layering, IRPs & I/O Stack Locations, IRP Processing, IRP Completion, IRP Data Buffering,

[1] Chapter 8: I/O Subsystem (I/O System Components, Device Drivers, I/O Processing)

Kernel Programming

Driver Development Environment

Windows Driver Kit, Building with Enterprise WDK, Targets, Platforms and Configurations, Kernel Debugging, Driver Symbols and Source Code, Driver Replacement Maps

Driver Programming Basics

Driver Entry Points, Windows Version APIs, WDK Headers, NTSTATUS Codes, Debug Prints, Memory Allocation, Unicode Strings.

[4] Chapter 6: Life in Kernel Space
[5] Chapter 3: Basic Programming Techniques

I/O Processing

User/Kernel Interface, Device Objects, Symbolic Links, Handling IRPs, Building IRPs.

[3] Chapter 2: Subverting the Kernel
[5] Chapter 5: The I/O Request Packet
[5] Chapter 9: I/O Control Operations

Queues & Serialization

Linked Lists, Waitable Locks, Spin Locks, Locking Granularity, Interlocked Operations.

[5] Chapter 4: Synchronization

Asynchronous Execution

DPC Routines, Kernel Timers, Worker Routines, Driver Threads, Thread Management.

Advanced Techniques

Locking and mapping memory, Rundown protection, Executive callbacks, Capturing stack back-traces, Registry Access, File Access.

Kernel Rootkits

Kernel Security Mitigations

NULL Page Allocation Prevention, Supervisor Mode Execution Protection, Safe Linking and Unlinking, Kernel Mode Code Signing, Kernel Patch Protection, Kernel Mode Data Execution Prevention, Non-Executable Pools, Kernel Address Space Layout Randomization,

Kernel Security Bypasses

Kernel Mode Shell Code Techniques, Data Execution Prevention Bypass, Kernel Mode Address Leaks, Kernel Mode Code Signing Bypass, Kernel Execution Vectors.

[4] Chapter 10: Building Shellcode in C

Hooking Techniques

Interception Types, Patch Guard, Inline Hooking, Memory Protection, Interrupt Hooking, Import Hooking, Dispatch Table Hooking

[3] Chapter 4: The Age-Old Art of Hooking
[3] Chapter 5: Runtime Patching
[4] Chapter 11: Modifying Call Tables
[4] Chapter 12: Modifying Code

Filtering Mechanisms

IRP Filter, Process & Thread Filter, Object Access Filter, Image Load Notification, Registry Callback, File System Mini-Filter, Early Load Anti-Malware (ELAM).

[3] Chapter 6: Layered Drivers
[5] Chapter 16: Filter Drivers

Covert Communications

Windows Filtering Platform, NDIS Intermediate Drivers, Net Buffer Lists & Net Buffers, NDIS Light Weight Filters, NDIS Internal Data Structures, NDIS Hooking.

[3] Chapter 9: Covert Channels
[4] Chapter 14: Covert Channels

Stealth Behavior

Process Attachment, Code Injection, Native APIs, Direct Kernel Object Manipulation, Stealth Operations, Self-Protection, Persistence and Startup.

[3] Chapter 7: Direct Kernel Object Manipulation

[4] Chapter 13: Modifying Kernel Objects