1.

Record Nr.	UNICAMPANIASUN0014847
Autore	De Stefano, Rodolfo
Titolo	1: Scritti di gnoseologia, scritti di filosofia prima / Rodolfo De Stefano
Pubbl/distr/stampa	Milano : Giuffre, 1997
ISBN	88-14-06500-4
Descrizione fisica	321 p. ; 24 cm.
Lingua di pubblicazione	Italiano
Formato	Materiale a stampa
Livello bibliografico	Monografia

2.

Record Nr.	UNINA9910825125303321
Autore	Dang Bruce
Titolo	Practical reverse engineering : x86, x64, ARM, Windows Kernel, reversing tools, and obfuscation / / Bruce Dang, Alexandre Gazet, Elias Bachaalany ; with contributions from Sebastien Josse
Pubbl/distr/stampa	Indianapolis, IN : , : John Wiley and Sons, , [2014] ©2014
ISBN	1-118-78739-0 1-118-78725-0
Edizione	[1st edition]
Descrizione fisica	1 online resource (383 p.)
Altri autori (Persone)	GazetAlexandre BachaalanyElias JosseSebastien
Disciplina	005.8
Soggetti	Reverse engineering
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Cover; Title Page; Copyright; Contents; Chapter 1 x86 and x64; Register Set and Data Types; Instruction Set; Syntax; Data Movement; Exercise; Arithmetic Operations; Stack Operations and Function

	Invocation; Exercises; Control Flow; System Mechanism; Address Translation; Interrupts and Exceptions; Walk-Through; Exercises; x64; Register Set and Data Types; Data Movement; Canonical Address; Function Invocation; Exercises; Chapter 2 ARM; Basic Features; Data Types and Registers; System-Level Controls and Settings; Introduction to the Instruction Set; Loading and Storing Data; LDR and STR Other Usage for LDRLDM and STM; PUSH and POP; Functions and Function Invocation; Arithmetic Operations; Branching and Conditional Execution; Thumb State; Switch-Case; Miscellaneous; Just-in-Time and Self-Modifying Code; Synchronization Primitives; System Services and Mechanisms; Instructions; Walk-Through; Next Steps; Exercises; Chapter 3 The Windows Kernel; Windows Fundamentals; Memory Layout; Processor Initialization; System Calls; Interrupt Request Level; Pool Memory; Memory Descriptor Lists; Processes and Threads; Execution Context; Kernel Synchronization Primitives; Lists Implementation Details Walk-Through; Exercises; Asynchronous and Ad-Hoc Execution; System Threads; Work Items; Asynchronous Procedure Calls; Deferred Procedure Calls; Timers; Process and Thread Callbacks; Completion Routines; I/O Request Packets; Structure of a Driver; Entry Points; Driver and Device Objects; IRP Handling; A Common Mechanism for User-Kernel Communication; Miscellaneous System Mechanisms; Walk-Throughs; An x86 Rootkit; An x64 Rootkit; Next Steps; Exercises; Building Confidence and Solidifying Your Knowledge; Investigating and Extending Your Knowledge Analysis of Real-Life Drivers Chapter 4 Debugging and Automation; The Debugging Tools and Basic Commands; Setting the Symbol Path; Debugger Windows; Evaluating Expressions; Process Control and Debut Events; Registers, Memory, and Symbols; Breakpoints; Inspecting Processes and Modules; Miscellaneous Commands; Scripting with the Debugging Tools; Pseudo-Registers; Aliases; Language; Script Files; Using Scripts Like Functions; Example Debug Scripts; Using the SDK; Concepts; Writing Debugging Tools Extensions; Useful Extensions, Tools, and Resources; Chapter 5 Obfuscation A Survey of Obfuscation Techniques The Nature of Obfuscation: A Motivating Example; Data-Based Obfuscations; Control-Based Obfuscation; Simultaneous Control-Flow and Data-Flow Obfuscation; Achieving Security by Obscurity; A Survey of Deobfuscation Techniques; The Nature of Deobfuscation: Transformation Inversion; Deobfuscation Tools; Practical Deobfuscation; Case Study; First Impressions; Analyzing Handlers Semantics; Symbolic Execution; Solving the Challenge; Final Thoughts; Exercises; Appendix Sample Names and Corresponding SHA1 Hashes; Index
Sommario/riassunto	Analyzing how hacks are done, so as to stop them in the future Reverse engineering is the process of analyzing hardware or software and understanding it, without having access to the source code or design documents. Hackers are able to reverse engineer systems and exploit what they find with scary results. Now the good guys can use the same tools to thwart these threats. Practical Reverse Engineering goes under the hood of reverse engineering for security analysts, security engineers, and system programmers, so they can learn how to use these same processes to stop hacke