

Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

A no-nonsense, practical guide to current and future processor and computer architectures, enabling you to design computer systems and develop better software applications across a variety of domains

Key Features

- Understand digital circuitry with the help of transistors, logic gates, and sequential logic*
- Examine the architecture and instruction sets of x86, x64, ARM, and RISC-V processors*
- Explore the architecture of modern devices such as the iPhone X and high-performance gaming PCs*

Book Description

Are you a software developer, systems designer, or computer architecture student looking for a methodical introduction to digital device architectures but overwhelmed by their complexity? This book will help you to learn how modern computer systems work, from the lowest level of transistor switching to the macro view of collaborating multiprocessor servers. You'll gain unique insights into the internal behavior of processors that execute the code developed in high-level languages and enable you to design more efficient and scalable software systems. The book will teach you the fundamentals of computer systems including transistors, logic gates, sequential logic, and instruction operations. You will learn details of modern processor architectures and instruction sets including x86, x64, ARM, and RISC-V. You will see how to implement a RISC-V processor in a low-cost FPGA board and how to write a quantum computing program and run it on an actual quantum computer. By the end of this book, you will have a thorough understanding of modern processor and computer architectures and the future directions these architectures are likely to take. What you will learn

Get to grips with transistor technology and digital circuit principles

- Discover the functional elements of computer processors*
- Understand pipelining and superscalar execution*
- Work with floating-point data formats*
- Understand the purpose and operation of the supervisor mode*
- Implement a complete RISC-V processor in a low-cost FPGA*
- Explore the techniques used in virtual machine implementation*
- Write a quantum computing program and run it on a quantum computer*

Who this book is for

This book is for software developers, computer engineering students, system designers, reverse engineers, and anyone looking to understand the architecture and design principles underlying modern computer systems from tiny embedded devices to warehouse-size cloud server farms. A general understanding of computer processors is helpful but not required.

; 0x40 assembly riddles "xchg rax,rax" is a collection of assembly gems and riddles I found over many years of reversing and writing assembly code. The book contains 0x40 short assembly snippets, each built to teach you one concept about assembly, math or life in general. Be warned - This book is not for beginners. It doesn't contain anything besides assembly code, and therefore some x86_64 assembly knowledge is required. How to use this book? Get an assembler (Yasm or Nasm is recommended), and obtain the x86_64 instruction set. Then for every snippet, try to understand what it does. Try to run it with different inputs if you don't understand it in the beginning. Look up for instructions you don't fully know in the Instruction sets PDF. Start from the beginning. The order has meaning. As a final note, the full contents of the book could be viewed for free on my website (Just google "xchg rax,rax").

Stop manually analyzing binary! Practical Binary Analysis is the first book of its kind to present advanced binary analysis topics, such as binary instrumentation, dynamic taint analysis, and symbolic execution, in an accessible way. As malware increasingly obfuscates itself and applies anti-analysis techniques to thwart our analysis, we need more sophisticated methods that allow us to raise that dark curtain designed to keep us out--binary analysis can help. The goal of all binary analysis is to determine (and possibly modify) the true properties of binary programs to understand what they really do, rather than what we think they should do. While reverse engineering and disassembly are critical first steps in many forms of binary analysis, there is much more to be learned. This hands-on guide teaches you how to tackle the fascinating but challenging topics of binary analysis and instrumentation and helps you become proficient in an area typically only mastered by a small group of expert hackers. It will take you from basic concepts to state-of-the-art methods as you dig into topics like code injection,

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

disassembly, dynamic taint analysis, and binary instrumentation. Written for security engineers, hackers, and those with a basic working knowledge of C/C++ and x86-64, Practical Binary Analysis will teach you in-depth how binary programs work and help you acquire the tools and techniques needed to gain more control and insight into binary programs. Once you've completed an introduction to basic binary formats, you'll learn how to analyze binaries using techniques like the GNU/Linux binary analysis toolchain, disassembly, and code injection. You'll then go on to implement profiling tools with Pin and learn how to build your own dynamic taint analysis tools with libdft and symbolic execution tools using Triton. You'll learn how to: - Parse ELF and PE binaries and build a binary loader with libbfd - Use data-flow analysis techniques like program tracing, slicing, and reaching definitions analysis to reason about runtime flow of your programs - Modify ELF binaries with techniques like parasitic code injection and hex editing - Build custom disassembly tools with Capstone - Use binary instrumentation to circumvent anti-analysis tricks commonly used by malware - Apply taint analysis to detect control hijacking and data leak attacks - Use symbolic execution to build automatic exploitation tools With exercises at the end of each chapter to help solidify your skills, you'll go from understanding basic assembly to performing some of the most sophisticated binary analysis and instrumentation. Practical Binary Analysis gives you what you need to work effectively with binary programs and transform your knowledge from basic understanding to expert-level proficiency.

Explore open-source Linux tools and advanced binary analysis techniques to analyze malware, identify vulnerabilities in code, and mitigate information security risks
Key Features
Adopt a methodological approach to binary ELF analysis on Linux Learn how to disassemble binaries and understand disassembled code Discover how and when to patch a malicious binary during analysis
Book Description
Binary analysis is the process of examining a binary program to determine information security actions. It is a complex, constantly evolving, and challenging topic that crosses over into several domains of information technology and security. This binary analysis book is designed to help you get started with the basics, before gradually advancing to challenging topics. Using a recipe-based approach, this book guides you through building a lab of virtual machines and installing tools to analyze binaries effectively. You'll begin by learning about the IA32 and ELF32 as well as IA64 and ELF64 specifications. The book will then guide you in developing a methodology and exploring a variety of tools for Linux binary analysis. As you advance, you'll learn how to analyze malicious 32-bit and 64-bit binaries and identify vulnerabilities. You'll even examine obfuscation and anti-analysis techniques, analyze polymorphed malicious binaries, and get a high-level overview of dynamic taint analysis and binary instrumentation concepts. By the end of the book, you'll have gained comprehensive insights into binary analysis concepts and have developed the foundational skills to confidently delve into the realm of binary analysis. What you will learn
Traverse the IA32, IA64, and ELF specifications Explore Linux tools to disassemble ELF binaries Identify vulnerabilities in 32-bit and 64-bit binaries Discover actionable solutions to overcome the limitations in analyzing ELF binaries Interpret the output of Linux tools to identify security risks in binaries Understand how dynamic taint analysis works
Who this book is for
This book is for anyone looking to learn how to dissect ELF binaries using open-source tools available in Linux. If you're a Linux system administrator or information security professional, you'll find this guide useful. Basic knowledge of Linux, familiarity with virtualization technologies and the working of network sockets, and experience in basic Python or Bash scripting will assist you with understanding the concepts in this book

A Guide to Kernel Exploitation: Attacking the Core discusses the theoretical techniques and approaches needed to develop reliable and effective kernel-level exploits, and applies them to different operating systems, namely, UNIX derivatives, Mac OS X, and Windows. Concepts and tactics are presented categorically so that even when a specifically detailed vulnerability has been patched, the foundational information provided will help hackers in writing a newer, better attack; or help pen testers, auditors, and the like develop a more concrete design and defensive structure. The book is organized into four parts. Part I introduces the kernel and sets out the theoretical basis on which to build the rest of the book. Part II focuses on different operating systems and describes exploits for them that target various bug classes. Part III on remote kernel exploitation analyzes the effects of the remote scenario and presents new techniques to target remote issues. It includes a step-by-step analysis of the development of a reliable, one-shot, remote exploit for a real vulnerability
a bug affecting the SCTP subsystem found in the

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

Linux kernel. Finally, Part IV wraps up the analysis on kernel exploitation and looks at what the future may hold. Covers a range of operating system families — UNIX derivatives, Mac OS X, Windows Details common scenarios such as generic memory corruption (stack overflow, heap overflow, etc.) issues, logical bugs and race conditions Delivers the reader from user-land exploitation to the world of kernel-land (OS) exploits/attacks, with a particular focus on the steps that lead to the creation of successful techniques, in order to give to the reader something more than just a set of tricks

Reverse Engineering: Mechanisms, Structures, Systems & Materials

Learning Malware Analysis

Subverting the Windows Kernel

Paleo Approach

Covers x86 64-bit, AVX, AVX2, and AVX-512

Practical Malware Analysis

Gray Hat Python

Gain the fundamentals of x86 64-bit assembly language programming and focus on the updated aspects of the x86 instruction set that are most relevant to application software development. This book covers topics including x86 64-bit programming and Advanced Vector Extensions (AVX) programming. The focus in this second edition is exclusively on 64-bit base programming architecture and AVX programming. Modern X86 Assembly Language Programming's structure and sample code are designed to help you quickly understand x86 assembly language programming and the computational capabilities of the x86 platform. After reading and using this book, you'll be able to code performance-enhancing functions and algorithms using x86 64-bit assembly language and the AVX, AVX2 and AVX-512 instruction set extensions. What You Will Learn Discover details of the x86 64-bit platform including its core architecture, data types, registers, memory addressing modes, and the basic instruction set Use the x86 64-bit instruction set to create performance-enhancing functions that are callable from a high-level language (C++) Employ x86 64-bit assembly language to efficiently manipulate common data types and programming constructs including integers, text strings, arrays, and structures Use the AVX instruction set to perform scalar floating-point arithmetic Exploit the AVX, AVX2, and AVX-512 instruction sets to significantly accelerate the performance of computationally-intense algorithms in problem domains such as image processing, computer graphics, mathematics, and statistics Apply various coding strategies and techniques to optimally exploit the x86 64-bit, AVX, AVX2, and AVX-512 instruction sets for maximum possible performance Who This Book Is For Software developers who want to learn how to write code using x86 64-bit assembly language. It's also ideal for software developers who already have a basic understanding of x86 32-bit or 64-bit assembly language programming and are interested in learning how to exploit the SIMD capabilities of AVX, AVX2 and AVX-512.

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

Beginning with a basic primer on reverse engineering-including computer internals, operating systems, and assembly language-and then discussing the various applications of reverse engineering, this book provides readers with practical, in-depth techniques for software reverse engineering. The book is broken into two parts, the first deals with security-related reverse engineering and the second explores the more practical aspects of reverse engineering. In addition, the author explains how to reverse engineer a third-party software library to improve interfacing and how to reverse engineer a competitor's software to build a better product. * The first popular book to show how software reverse engineering can help defend against security threats, speed up development, and unlock the secrets of competitive products * Helps developers plug security holes by demonstrating how hackers exploit reverse engineering techniques to crack copy-protection schemes and identify software targets for viruses and other malware * Offers a primer on advanced reverse-engineering, delving into "disassembly"-code-level reverse engineering-and explaining how to decipher assembly language

Malware analysis is big business, and attacks can cost a company dearly. When malware breaches your defenses, you need to act quickly to cure current infections and prevent future ones from occurring. For those who want to stay ahead of the latest malware, Practical Malware Analysis will teach you the tools and techniques used by professional analysts. With this book as your guide, you'll be able to safely analyze, debug, and disassemble any malicious software that comes your way. You'll learn how to: –Set up a safe virtual environment to analyze malware –Quickly extract network signatures and host-based indicators –Use key analysis tools like IDA Pro, OllyDbg, and WinDbg –Overcome malware tricks like obfuscation, anti-disassembly, anti-debugging, and anti-virtual machine techniques –Use your newfound knowledge of Windows internals for malware analysis –Develop a methodology for unpacking malware and get practical experience with five of the most popular packers –Analyze special cases of malware with shellcode, C++, and 64-bit code Hands-on labs throughout the book challenge you to practice and synthesize your skills as you dissect real malware samples, and pages of detailed dissections offer an over-the-shoulder look at how the pros do it. You'll learn how to crack open malware to see how it really works, determine what damage it has done, thoroughly clean your network, and ensure that the malware never comes back. Malware analysis is a cat-and-mouse game with rules that are constantly changing, so make sure you have the fundamentals. Whether you're tasked with securing one network or a thousand networks, or you're making a living as a malware analyst, you'll find what you need to succeed in Practical Malware Analysis.

The Batchography book is a boon for system administrators, build engineers, programmers and home users alike. It takes you on a journey of re-discovery of the lost art of Batch files programming. Whether you are an experienced user or new to the language, you will be surprised by the clarity and the abundance of the material presented in this book. With more

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

than 140 scripting recipes, you will learn about things that you never thought were possible to achieve using the Batch files scripting language.

Rootkits and Bootkits will teach you how to understand and counter sophisticated, advanced threats buried deep in a machine's boot process or UEFI firmware. With the aid of numerous case studies and professional research from three of the world's leading security experts, you'll trace malware development over time from rootkits like TDL3 to present-day UEFI implants and examine how they infect a system, persist through reboot, and evade security software. As you inspect and dissect real malware, you'll learn:

- How Windows boots—including 32-bit, 64-bit, and UEFI mode—and where to find vulnerabilities
- The details of boot process security mechanisms like Secure Boot, including an overview of Virtual Secure Mode (VSM) and Device Guard
- Reverse engineering and forensic techniques for analyzing real malware, including bootkits like Rovnix/Carberp, Gapz, TDL4, and the infamous rootkits TDL3 and Festi
- How to perform static and dynamic analysis using emulation and tools like Bochs and IDA Pro
- How to better understand the delivery stage of threats against BIOS and UEFI firmware in order to create detection capabilities
- How to use virtualization tools like VMware Workstation to reverse engineer bootkits and the Intel Chipsec tool to dig into forensic analysis

Cybercrime syndicates and malicious actors will continue to write ever more persistent and covert attacks, but the game is not lost. Explore the cutting edge of malware analysis with Rootkits and Bootkits. Covers boot processes for Windows 32-bit and 64-bit operating systems.

Reverse-Engineering Nature

Malware Analyst's Cookbook and DVD

Trends in Computer Science, Engineering and Information Technology

Go Programming For Hackers and Pentesters

Python Programming for Hackers and Reverse Engineers

Penetration Testing with Shellcode

Unleash Kali Linux, PowerShell, and Windows debugging tools for security testing and analysis

The First In-Depth, Real-World, Insider's Guide to Powerful Windows Debugging For Windows developers, few tasks are more challenging than debugging--or more crucial. Reliable and realistic information about Windows debugging has always been scarce. Now, with over 15 years of experience two of Microsoft's system-level developers present a thorough and practical guide to Windows debugging ever written. Mario Hewardt and Daniel Pravat cover debugging throughout the entire application lifecycle and show how to make the most of

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

the tools currently available--including Microsoft's powerful native debuggers and third-party solutions. To help you find real solutions fast, this book is organized around real-world debugging scenarios. Hewardt and Pravat use detailed code examples to illuminate the complex debugging challenges professional developers actually face. From core Windows operating system concepts to security, Windows® Vista™ and 64-bit debugging, they address emerging topics head-on-and nothing is ever oversimplified or glossed over!

Like the best-selling Black Hat Python, Black Hat Go explores the darker side of the popular Go programming language. This collection of short scripts will help you test your systems, build and automate tools to fit your needs, and improve your offensive security skillset. Black Hat Go explores the darker side of Go, the popular programming language revered by hackers for its simplicity, efficiency, and reliability. It provides an arsenal of practical tactics from the perspective of security practitioners and hackers to help you test your systems, build and automate tools to fit your needs, and improve your offensive security skillset, all using the power of Go. You'll begin your journey with a basic overview of Go's syntax and philosophy and then start to explore examples that you can leverage for tool development, including common network protocols like HTTP, DNS, and SMB. You'll then dig into various tactics and problems that penetration testers encounter, addressing things like data pilfering, packet sniffing, and exploit development. You'll create dynamic, pluggable tools before diving into cryptography, attacking Microsoft Windows, and implementing steganography. You'll learn how to:

- Make performant tools that can be used for your own security projects*
- Create usable tools that interact with remote APIs*
- Scrape arbitrary HTML data*
- Use Go's standard package, net/http, for building HTTP servers*
- Write your own DNS server and proxy*
- Use DNS tunneling to establish a C2 channel out of a restrictive network*
- Create a vulnerability fuzzer to discover an application's security weaknesses*
- Use plug-ins and extensions to future-proof products*
- Build an RC2 symmetric-key brute-forcer*
- Implant data within a Portable Network Graphics (PNG) image.*

Are you ready to add to your arsenal of security tools? Then let's Go!

Analyzing how hacks are done, so as to stop them in the future Reverse engineering is the

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

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 hackers in their tracks. The book covers x86, x64, and ARM (the first book to cover all three); Windows kernel-mode code rootkits and drivers; virtual machine protection techniques; and much more. Best of all, it offers a systematic approach to the material, with plenty of hands-on exercises and real-world examples. Offers a systematic approach to understanding reverse engineering, with hands-on exercises and real-world examples. Covers x86, x64, and advanced RISC machine (ARM) architectures as well as deobfuscation and virtual machine protection techniques. Provides special coverage of Windows kernel-mode code (rootkits/drivers), a topic not often covered elsewhere, and explains how to analyze drivers step by step. Demystifies topics that have a steep learning curve. Includes a bonus chapter on reverse engineering tools. Practical Reverse Engineering: Using x86, x64, ARM, Windows Kernel, and Reversing Tools provides crucial, up-to-date guidance for a broad range of IT professionals.

If you want to master the art and science of reverse engineering code with IDA Pro for security R&D or software debugging, this is the book for you. Highly organized and sophisticated criminal entities are constantly developing more complex, obfuscated, and armored viruses, worms, Trojans, and botnets. IDA Pro's interactive interface and programmable development language provide you with complete control over code disassembly and debugging. This is the only book which focuses exclusively on the world's most powerful and popular tool for reverse engineering code. *Reverse Engineer REAL Hostile Code To follow along with this chapter, you must download a file called !DANGER!INFECTEDMALWARE!DANGER!... 'nuff said. *Portable Executable (PE) and Executable and Linking Formats (ELF) Understand the physical layout of PE and ELF files, and analyze the components that are essential to reverse engineering. *Break Hostile Code Armor and Write your own Exploits Understand execution flow, trace functions, recover hard coded

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

*passwords, find vulnerable functions, backtrace execution, and craft a buffer overflow. *Master Debugging Debug in IDA Pro, use a debugger while reverse engineering, perform heap and stack access modification, and use other debuggers. *Stop Anti-Reversing Anti-reversing, like reverse engineering or coding in assembly, is an art form. The trick of course is to try to stop the person reversing the application. Find out how! *Track a Protocol through a Binary and Recover its Message Structure Trace execution flow from a read event, determine the structure of a protocol, determine if the protocol has any undocumented messages, and use IDA Pro to determine the functions that process a particular message. *Develop IDA Scripts and Plug-ins Learn the basics of IDA scripting and syntax, and write IDC scripts and plug-ins to automate even the most complex tasks. This much-anticipated revision, written by the ultimate group of top security experts in the world, features 40 percent new content on how to find security holes in any operating system or application New material addresses the many new exploitation techniques that have been discovered since the first edition, including attacking "unbreakable" software packages such as McAfee's Enterscept, Mac OS X, XP, Office 2003, and Vista Also features the first-ever published information on exploiting Cisco's IOS, with content that has never before been explored The companion Web site features downloadable code files System architecture, processes, threads, memory management, and more*

Xchg Rax, Rax

A Guide to Kernel Exploitation

The Hands-On Guide to Dissecting Malicious Software

Re-engineer your ethical hacking skills

The Shellcoder's Handbook

Binary Analysis Cookbook

Master the art of identifying vulnerabilities within the Windows OS and develop the desired solutions for it using Kali Linux.

Key Features Identify the vulnerabilities in your system using Kali Linux 2018.02 Discover the art of exploiting Windows

kernel drivers Get to know several bypassing techniques to gain control of your Windows environment Book Description

Windows has always been the go-to platform for users around the globe to perform administration and ad hoc tasks, in settings that range from small offices to global enterprises, and this massive footprint makes securing Windows a unique challenge.

This book will enable you to distinguish yourself to your clients. In this book, you'll learn advanced techniques to attack

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

Windows environments from the indispensable toolkit that is Kali Linux. We'll work through core network hacking concepts and advanced Windows exploitation techniques, such as stack and heap overflows, precision heap spraying, and kernel exploitation, using coding principles that allow you to leverage powerful Python scripts and shellcode. We'll wrap up with post-exploitation strategies that enable you to go deeper and keep your access. Finally, we'll introduce kernel hacking fundamentals and fuzzing testing, so you can discover vulnerabilities and write custom exploits. By the end of this book, you'll be well-versed in identifying vulnerabilities within the Windows OS and developing the desired solutions for them. What you will learn Get to know advanced pen testing techniques with Kali Linux Gain an understanding of Kali Linux tools and methods from behind the scenes See how to use Kali Linux at an advanced level Understand the exploitation of Windows kernel drivers Understand advanced Windows concepts and protections, and how to bypass them using Kali Linux Discover Windows exploitation techniques, such as stack and heap overflows and kernel exploitation, through coding principles Who this book is for This book is for penetration testers, ethical hackers, and individuals breaking into the pentesting role after demonstrating an advanced skill in boot camps. Prior experience with Windows exploitation, Kali Linux, and some Windows debugging tools is necessary Memory forensics provides cutting edge technology to help investigate digital attacks Memory forensics is the art of analyzing computer memory (RAM) to solve digital crimes. As a follow-up to the best seller Malware Analyst's Cookbook, experts in the fields of malware, security, and digital forensics bring you a step-by-step guide to memory forensics—now the most sought after skill in the digital forensics and incident response fields. Beginning with introductory concepts and moving toward the advanced, The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory is based on a five day training course that the authors have presented to hundreds of students. It is the only book on the market that focuses exclusively on memory forensics and how to deploy such techniques properly. Discover memory forensics techniques: How volatile memory analysis improves digital investigations Proper investigative steps for detecting stealth malware and advanced threats How to use free, open source tools for conducting thorough memory forensics Ways to acquire memory from suspect systems in a forensically sound manner The next era of malware and security breaches are more sophisticated and targeted, and the volatile memory of a computer is often overlooked or destroyed as part of the incident response process. The Art of Memory Forensics explains the latest technological innovations in digital forensics to help bridge this gap. It covers the most popular and recently released versions of Windows, Linux, and Mac, including both the 32 and 64-bit editions. Practical Reverse Engineeringx86, x64, ARM, Windows Kernel, Reversing Tools, and ObfuscationJohn Wiley & Sons Hack your antivirus software to stamp out future vulnerabilities The Antivirus Hacker's Handbook guides you through the process of reverse engineering antivirus software. You explore how to detect and exploit vulnerabilities that can be leveraged to improve future software design, protect your network, and anticipate attacks that may sneak through your antivirus' line of defense. You'll begin building your knowledge by diving into the reverse engineering process, which details how to start from a finished antivirus software program and work your way back through its development using the functions and other key elements of the software. Next, you leverage your new knowledge about software development to evade, attack, and exploit antivirus software—all of which can help you strengthen your network and protect your data. While not all viruses are damaging, understanding how to better protect your computer against them can help you maintain the integrity of your network. Discover

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

how to reverse engineer your antivirus software Explore methods of antivirus software evasion Consider different ways to attack and exploit antivirus software Understand the current state of the antivirus software market, and get recommendations for users and vendors who are leveraging this software The Antivirus Hacker's Handbook is the essential reference for software reverse engineers, penetration testers, security researchers, exploit writers, antivirus vendors, and software engineers who want to understand how to leverage current antivirus software to improve future applications. Master malware analysis to protect your systems from getting infected Key Features Set up and model solutions, investigate malware, and prevent it from occurring in future Learn core concepts of dynamic malware analysis, memory forensics, decryption, and much more A practical guide to developing innovative solutions to numerous malware incidents Book Description With the ever-growing proliferation of technology, the risk of encountering malicious code or malware has also increased. Malware analysis has become one of the most trending topics in businesses in recent years due to multiple prominent ransomware attacks. Mastering Malware Analysis explains the universal patterns behind different malicious software types and how to analyze them using a variety of approaches. You will learn how to examine malware code and determine the damage it can possibly cause to your systems to ensure that it won't propagate any further. Moving forward, you will cover all aspects of malware analysis for the Windows platform in detail. Next, you will get to grips with obfuscation and anti-disassembly, anti-debugging, as well as anti-virtual machine techniques. This book will help you deal with modern cross-platform malware. Throughout the course of this book, you will explore real-world examples of static and dynamic malware analysis, unpacking and decrypting, and rootkit detection. Finally, this book will help you strengthen your defenses and prevent malware breaches for IoT devices and mobile platforms. By the end of this book, you will have learned to effectively analyze, investigate, and build innovative solutions to handle any malware incidents. What you will learn Explore widely used assembly languages to strengthen your reverse-engineering skills Master different executable file formats, programming languages, and relevant APIs used by attackers Perform static and dynamic analysis for multiple platforms and file types Get to grips with handling sophisticated malware cases Understand real advanced attacks, covering all stages from infiltration to hacking the system Learn to bypass anti-reverse engineering techniques Who this book is for If you are an IT security administrator, forensic analyst, or malware researcher looking to secure against malicious software or investigate malicious code, this book is for you. Prior programming experience and a fair understanding of malware attacks and investigation is expected.

The complete malware analyst's guide to combating malicious software, APT, cybercrime, and IoT attacks

Practical Reverse Engineering

Developing Autonomous Bots for Online Games

Reversing Modern Malware and Next Generation Threats

Hands-On Penetration Testing on Windows

Batchography

Windows Internals, Part 1

You don't need to be a wizard to transform a game you like into a game you love. Imagine if you could give your favorite PC game a more informative heads-up display or instantly collect all that loot from your latest epic battle. Bring your knowledge of Windows-based development

and memory management, and Game Hacking will teach you what you need to become a true game hacker. Learn the basics, like reverse engineering, assembly code analysis, programmatic memory manipulation, and code injection, and hone your new skills with hands-on example code and practice binaries. Level up as you learn how to: –Scan and modify memory with Cheat Engine –Explore program structure and execution flow with OllyDbg –Log processes and pinpoint useful data files with Process Monitor –Manipulate control flow through NOPing, hooking, and more –Locate and dissect common game memory structures You’ll even discover the secrets behind common game bots, including: –Extrasensory perception hacks, such as wallhacks and heads-up displays –Responsive hacks, such as autohealers and combo bots –Bots with artificial intelligence, such as cave walkers and automatic looters Game hacking might seem like black magic, but it doesn’t have to be. Once you understand how bots are made, you’ll be better positioned to defend against them in your own games. Journey through the inner workings of PC games with Game Hacking, and leave with a deeper understanding of both game design and computer security.

More practical less theory KEY FEATURES ? In-depth practical demonstration with multiple examples of reverse engineering concepts. ? Provides a step-by-step approach to reverse engineering, including assembly instructions. ? Helps security researchers to crack application code and logic using reverse engineering open source tools. ? Reverse engineering strategies for simple-to-complex applications like Wannacry ransomware and Windows calculator. **DESCRIPTION** The book ‘Implementing Reverse Engineering’ begins with a step-by-step explanation of the fundamentals of reverse engineering. You will learn how to use reverse engineering to find bugs and hacks in real-world applications. This book is divided into three sections. The first section is an exploration of the reverse engineering process. The second section explains reverse engineering of applications, and the third section is a collection of real-world use-cases with solutions. The first section introduces the basic concepts of a computing system and the data building blocks of the computing system. This section also includes open-source tools such as CFF Explorer, Ghidra, Cutter, and x32dbg. The second section goes over various reverse engineering practicals on various applications to give users hands-on experience. In the third section, reverse engineering of Wannacry ransomware, a well-known Windows application, and various exercises are demonstrated step by step. In a very detailed and step-by-step manner, you will practice and understand different assembly instructions, types of code calling conventions, assembly patterns of applications with the printf function, pointers, array, structure, scanf, strcpy function, decision, and loop control structures. You will learn how to use open-source tools for reverse engineering such as portable executable editors, disassemblers, and debuggers. **WHAT YOU WILL LEARN ?** Understand different code calling conventions like CDECL, STDCALL, and FASTCALL with practical illustrations. ? Analyze and break WannaCry ransomware using Ghidra. ? Using Cutter, reconstruct application logic from the assembly code. ? Hack the Windows calculator to modify its behavior. **WHO THIS BOOK IS FOR** This book is for cybersecurity researchers, bug bounty hunters, software developers, software testers, and software quality assurance experts who want to perform reverse engineering for advanced security from attacks. Interested readers can also be from high schools or universities (with a Computer Science background). Basic programming knowledge is helpful but not required. **TABLE OF CONTENTS** 1. Impact of Reverse Engineering 2. Understanding Architecture of x86 machines 3. Up and Running with Reverse Engineering tools 4. Walkthrough on Assembly Instructions 5. Types of Code Calling Conventions 6. Reverse Engineering Pattern of Basic Code 7. Reverse Engineering Pattern of the printf() Program 8. Reverse Engineering Pattern of the Pointer Program 9. Reverse Engineering Pattern of the Decision Control Structure 10. Reverse Engineering Pattern of the Loop Control Structure 11. Array Code Pattern in Reverse Engineering 12. Structure Code Pattern in Reverse Engineering 13. Scanf Program Pattern in Reverse Engineering 14. strcpy Program Pattern in Reverse Engineering 15. Simple Interest Code Pattern in Reverse Engineering 16. Breaking Wannacry Ransomware with Reverse Engineering 17. Generate Pseudo Code from the Binary File 18. Fun with

Windows Calculator Using Reverse Engineering

A comprehensive look at reverse engineering as a legitimate learning, design, and troubleshooting tool This unique book examines the often underappreciated and occasionally maligned technique of reverse engineering. More than a shortcut for the lazy or unimaginative to reproduce an artless copy of an existing creation, reverse engineering is an essential brick – if not a keystone – in the pathway to a society’s technological advancement. Written by an engineer who began teaching after years in industry, Reverse Engineering reviews this meticulous analytical process with a breadth and depth as never before. Find out how to: Learn by “mechanical dissection” Deduce the role, purpose, and functionality of a designed entity Identify materials-of-construction and methods-of-manufacture by observation alone Assess the suitability of a design to purpose from form and fit The rich heritage of engineering breakthroughs enabled by reverse engineering is also discussed. This is not a dry textbook. It is the engaging and enlightening account of the journey of engineering from the astounding creations of ancient cultures to what, with the aid of reverse engineering, promises to be an even more astounding future! Coverage includes: Methods of product teardown Failure analysis and forensic engineering Deducing or inferring role, purpose, and functionality during reverse engineering The Antikythera mechanism Identifying materials-of-construction Inferring methods-of-manufacture or -construction Construction of Khufu’s pyramid Assessing design suitability Value and production engineering Reverse engineering of materials and substances Reverse engineering of broken, worn, or obsolete parts for remanufacture The law and the ethics of reverse engineering

A guide to using the Ghidra software reverse engineering tool suite. The result of more than a decade of research and development within the NSA, the Ghidra platform was developed to address some of the agency's most challenging reverse-engineering problems. With the open-source release of this formerly restricted tool suite, one of the world's most capable disassemblers and intuitive decompilers is now in the hands of cybersecurity defenders everywhere -- and The Ghidra Book is the one and only guide you need to master it. In addition to discussing RE techniques useful in analyzing software and malware of all kinds, the book thoroughly introduces Ghidra's components, features, and unique capacity for group collaboration. You'll learn how to: • Navigate a disassembly • Use Ghidra's built-in decompiler to expedite analysis • Analyze obfuscated binaries • Extend Ghidra to recognize new data types • Build new Ghidra analyzers and loaders • Add support for new processors and instruction sets • Script Ghidra tasks to automate workflows • Set up and use a collaborative reverse engineering environment Designed for beginner and advanced users alike, The Ghidra Book will effectively prepare you to meet the needs and challenges of RE, so you can analyze files like a pro.

A guide to rootkits describes what they are, how they work, how to build them, and how to detect them.

The Art of Batch Files Programming

Detecting Malware and Threats in Windows, Linux, and Mac Memory

Explore the concepts, tools, and techniques to analyze and investigate Windows malware

Actionable Recipes for Disassembling and Analyzing Binaries for Security Risks

Mastering Reverse Engineering

Technology of Reinvention

Rootkits

This book creates a framework for understanding and using cyberpower in support of

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

national security. Cyberspace and cyberpower are now critical elements of international security. United States needs a national policy which employs cyberpower to support its national security interests.

Understand malware analysis and its practical implementation Key Features Explore the key concepts of malware analysis and memory forensics using real-world examples Learn the art of detecting, analyzing, and investigating malware threats Understand adversary tactics and techniques Book Description Malware analysis and memory forensics are powerful analysis and investigation techniques used in reverse engineering, digital forensics, and incident response. With adversaries becoming sophisticated and carrying out advanced malware attacks on critical infrastructures, data centers, and private and public organizations, detecting, responding to, and investigating such intrusions is critical to information security professionals. Malware analysis and memory forensics have become must-have skills to fight advanced malware, targeted attacks, and security breaches. This book teaches you the concepts, techniques, and tools to understand the behavior and characteristics of malware through malware analysis. It also teaches you techniques to investigate and hunt malware using memory forensics. This book introduces you to the basics of malware analysis, and then gradually progresses into the more advanced concepts of code analysis and memory forensics. It uses real-world malware samples, infected memory images, and visual diagrams to help you gain a better understanding of the subject and to equip you with the skills required to analyze, investigate, and respond to malware-related incidents. What you will learn Create a safe and isolated lab environment for malware analysis Extract the metadata associated with malware Determine malware's interaction with the system Perform code analysis using IDA Pro and x64dbg Reverse-engineer various malware functionalities Reverse engineer and decode common encoding/encryption algorithms Reverse-engineer malware code injection and hooking techniques Investigate and hunt malware using memory forensics Who this book is for This book is for incident responders, cyber-security investigators, system administrators, malware analyst, forensic practitioners, student, or curious security professionals interested in learning malware analysis and memory forensics. Knowledge of programming

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

languages such as C and Python is helpful but is not mandatory. If you have written few lines of code and have a basic understanding of programming concepts, you'll be able to get most out of this book.

Python is fast becoming the programming language of choice for hackers, reverse engineers, and software testers because it's easy to write quickly, and it has the low-level support and libraries that make hackers happy. But until now, there has been no real manual on how to use Python for a variety of hacking tasks. You had to dig through forum posts and man pages, endlessly tweaking your own code to get everything working. Not anymore. Gray Hat Python explains the concepts behind hacking tools and techniques like debuggers, trojans, fuzzers, and emulators. But author Justin Seitz goes beyond theory, showing you how to harness existing Python-based security tools—and how to build your own when the pre-built ones won't cut it. You'll learn how to:

- Automate tedious reversing and security tasks
- Design and program your own debugger
- Learn how to fuzz Windows drivers and create powerful fuzzers from scratch
- Have fun with code and library injection, soft and hard hooking techniques, and other software trickery
- Sniff secure traffic out of an encrypted web browser session
- Use PyDBG, Immunity Debugger, Sulley, IDAPython, PyEMU, and more

The world's best hackers are using Python to do their handiwork. Shouldn't you?

This book constitutes the refereed proceedings of the First International Conference on Computer Science, Engineering and Information Technology, CCSEIT 2011, held in Tirunelveli, India, in September 2011. The 73 revised full papers were carefully reviewed and selected from more than 400 initial submissions. The papers feature significant contributions to all major fields of the Computer Science and Information Technology in theoretical and practical aspects.

No source code? No problem. With IDA Pro, the interactive disassembler, you live in a source code-optional world. IDA can automatically analyze the millions of opcodes that make up an executable and present you with a disassembly. But at that point, your work is just beginning. With The IDA Pro Book, you'll learn how to turn that mountain of mnemonics into something you can actually use. Hailed by the creator of IDA Pro as

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

"profound, comprehensive, and accurate," the second edition of The IDA Pro Book covers everything from the very first steps to advanced automation techniques. You'll find complete coverage of IDA's new Qt-based user interface, as well as increased coverage of the IDA debugger, the Bochs debugger, and IDA scripting (especially using IDAPython). But because humans are still smarter than computers, you'll even learn how to use IDA's latest interactive and scriptable interfaces to your advantage. Save time and effort as you learn to: -Navigate, comment, and modify disassembly -Identify known library routines, so you can focus your analysis on other areas of the code -Use code graphing to quickly make sense of cross references and function calls -Extend IDA to support new processors and filetypes using the SDK -Explore popular plug-ins that make writing IDA scripts easier, allow collaborative reverse engineering, and much more -Use IDA's built-in debugger to tackle hostile and obfuscated code Whether you're analyzing malware, conducting vulnerability research, or reverse engineering software, a mastery of IDA is crucial to your success. Take your skills to the next level with this 2nd edition of The IDA Pro Book.

Discovering and Exploiting Security Holes

Advanced Windows Debugging

First International Conference, CCSEIT 2011, Tirunelveli, Tamil Nadu, India, September 23-25, 2011, Proceedings

Essential MATLAB for Scientists and Engineers

The Definitive Guide

Secrets of Reverse Engineering

1. REVERSE OSMOSIS BASIC CONCEPTS - 2. FEED WATER TYPE AND ANALYSIS - 3. RAW WATER REQUIREMENTS - 4. SEA WATER INTAKE - 5. SEA WATER DOSING SYSTEMS - 6. REVERSE OSMOSIS PRETREATMENT CONVENTIONAL PRETREATMENT - 7. REVERSE OSMOSIS PRETREATMENT MICROFILTRATION and ULTRAFILTRATION - 8. MATERIALS - 9. REVERSE OSMOSIS MEMBRANES - 10. PRESSURE VESSELS AND RACKS - 11. REVERSE OSMOSIS PUMPS - 12. RECOVERY SYSTEMS - 13. REVERSE OSMOSIS RACKS CONTROL - 14. REVERSE OSMOSIS RACKS EQUIPMENT - 15. RACKS CLEANING SYSTEM and FLUSHING - 16. TREATED WATER CONDITIONING - 17. TREATED WATER DEPOSIT AND PUMPING - 18. NEUTRALIZATION, EFFLUENTS TREATMENT AND BRINE DISCHARGE - 19. ELECTRICAL EQUIPMENT - 20. CONTROL SYSTEMS - 21. VARIOUS EQUIPMENT - 22.

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

COST EVALUATION OF DESALINATION PLANTS - BISAC: 1: TEC005050 Technology & Engineering : Construction - HVAC 2: TEC009070 Technology & Engineering : Mechanical 3: TEC010030 Technology & Engineering : Environmental - Water Supply

While forensic analysis has proven to be a valuable investigative tool in the field of computer security, utilizing anti-forensic technology makes it possible to maintain a covert operational foothold for extended periods, even in a high-security environment. Adopting an approach that favors full disclosure, the updated Second Edition of The Rootkit Arsenal presents the most accessible, timely, and complete coverage of forensic countermeasures. This book covers more topics, in greater depth, than any other currently available. In doing so the author forges through the murky back alleys of the Internet, shedding light on material that has traditionally been poorly documented, partially documented, or intentionally undocumented. The range of topics presented includes how to: -Evade post-mortem analysis -Frustrate attempts to reverse engineer your command & control modules -Defeat live incident response -Undermine the process of memory analysis -Modify subsystem internals to feed misinformation to the outside -Entrench your code in fortified regions of execution -Design and implement covert channels -Unearth new avenues of attack

Based on a teach-yourself approach, the fundamentals of MATLAB are illustrated throughout with many examples from a number of different scientific and engineering areas, such as simulation, population modelling, and numerical methods, as well as from business and everyday life. Some of the examples draw on first-year university level maths, but these are self-contained so that their omission will not detract from learning the principles of using MATLAB. This completely revised new edition is based on the latest version of MATLAB. New chapters cover handle graphics, graphical user interfaces (GUIs), structures and cell arrays, and importing/exporting data. The chapter on numerical methods now includes a general GUI-driver ODE solver. * Maintains the easy informal style of the first edition * Teaches the basic principles of scientific programming with MATLAB as the vehicle * Covers the latest version of MATLAB

The process of reverse engineering has proven infinitely useful for analyzing Original Equipment Manufacturer (OEM) components to duplicate or repair them, or simply improve on their design. A guidebook to the rapid-fire changes in this area, Reverse Engineering: Technology of Reinvention introduces the fundamental principles, advanced methodologies, and other essential aspects of reverse engineering. The book's primary objective is twofold: to advance the technology of reinvention through reverse engineering and to improve the competitiveness of commercial parts in the aftermarket. Assembling and synergizing material from several different fields, this book prepares readers with the skills, knowledge, and abilities required to successfully apply reverse engineering in diverse fields ranging from aerospace, automotive, and medical device industries to academic research, accident investigation, and legal and forensic analyses. With this mission of preparation in mind, the author offers real-world examples to: Enrich readers' understanding of reverse engineering processes, empowering them with alternative options regarding part production Explain the latest technologies, practices, specifications, and regulations in reverse engineering Enable readers to judge if a "duplicated or repaired" part will meet the design functionality of the OEM part This book sets itself apart by covering seven key subjects: geometric measurement, part evaluation, materials identification, manufacturing process verification, data analysis, system compatibility, and intelligent property protection. Helpful in making new, compatible products that are cheaper than others on the market, the author provides the tools to uncover or clarify features of commercial products that were either previously unknown, misunderstood, or not used in the most effective way.

An estimated 50 million Americans suffer from some form of autoimmune disease. If you're among them, you may know all too well how little modern medicine can do to alleviate your condition. But that's no reason to give up hope. In this groundbreaking book, Sarah D. Ballantyne,

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

Ph.D., draws upon current medical research and her own battle with an autoimmune disorder to show you how you can become completely symptom-free—the natural way. The Paleo Approach is the first book ever to explain how to adapt the Paleo diet and lifestyle to bring about a full recovery. Read it to learn why foods marketed as "healthy"—such as whole grains, soy, and low-fat dairy—can contribute to the development of autoimmune conditions. Discover what you can eat to calm your immune system, reduce inflammation, and help your body heal itself. Find out which simple lifestyle changes—along with changes in diet—will make the biggest difference for your health. There's no need to worry that "going Paleo" will break the bank or require too much time in the kitchen preparing special foods. In The Paleo Approach, Dr. Ballantyne provides expert tips on how to make the switch easily and economically. Complete food lists with strategies for the day-to-day—how stay within your food budget, where to shop for what you need, how to make the most out of your time in the kitchen, and how to eat out—take all the guesswork out of going Paleo. Simple strategies for lifestyle adjustments, including small steps that can make a huge difference, guide you through the most important changes to support healing. Do you have a complicated condition that requires medical intervention, medication, or supplements? Dr. Ballantyne also walks you through the most useful medical tests, treatments, and supplements (as well as the most counterproductive ones) to help you open a dialogue with your physician. Features such as these make The Paleo Approach the ultimate resource for anyone suffering from an autoimmune disease. Why suffer a moment longer? Reclaim your health with The Paleo Approach!

The IDA Pro Book, 2nd Edition

Tools and Techniques for Fighting Malicious Code

Learn x86, ARM, and RISC-V architectures and the design of smartphones, PCs, and cloud servers

Implementing Reverse Engineering

Escape and Evasion in the Dark Corners of the System

Analyze, identify, and avoid malicious code and potential threats in your networks and systems

Cyberpower and National Security

Implement reverse engineering techniques to analyze software, exploit software targets, and defend against security threats like malware and viruses. Key Features Analyze and improvise software and hardware with real-world examples Learn advanced debugging and patching techniques with tools such as IDA Pro, x86dbg, and Radare2. Explore modern security techniques to identify, exploit, and avoid cyber threats **Book Description** If you want to analyze software in order to exploit its weaknesses and strengthen its defenses, then you should explore reverse engineering. Reverse Engineering is a hackerfriendly tool used to expose security flaws and questionable privacy practices. In this book, you will learn how to analyse software even without having access to its source code or design documents. You will start off by learning the low-level language used to communicate with the computer and then move on to covering reverse engineering techniques. Next, you will explore analysis techniques using real-world tools such as IDA Pro and x86dbg. As you progress through the chapters, you will walk through use cases encountered in reverse engineering, such as encryption and compression, used to obfuscate code, and how to to identify and overcome anti-debugging and anti-analysis tricks. Lastly, you will

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

learn how to analyse other types of files that contain code. By the end of this book, you will have the confidence to perform reverse engineering. What you will learn

Learn core reverse engineering

Identify and extract malware components

Explore the tools used for reverse engineering

Run programs under non-native operating systems

Understand binary obfuscation techniques

Identify and analyze anti-debugging and anti-analysis tricks

Who this book is for

If you are a security engineer or analyst or a system programmer and want to use reverse engineering to improve your software and hardware, this is the book for you. You will also find this book useful if you are a developer who wants to explore and learn reverse engineering. Having some programming/shell scripting knowledge is an added advantage.

Detect potential bugs in your code or program and develop your own tools using the Ghidra reverse engineering framework developed by the NSA project

Key Features

Make the most of Ghidra on different platforms such as Linux, Windows, and macOS

Leverage a variety of plug-ins and extensions to perform disassembly, assembly, decompilation, and scripting

Discover how you can meet your cybersecurity needs by creating custom patches and tools

Book Description

Ghidra, an open source software reverse engineering (SRE) framework created by the NSA research directorate, enables users to analyze compiled code on any platform, whether Linux, Windows, or macOS. This book is a starting point for developers interested in leveraging Ghidra to create patches and extend tool capabilities to meet their cybersecurity needs. You'll begin by installing Ghidra and exploring its features, and gradually learn how to automate reverse engineering tasks using Ghidra plug-ins. You'll then see how to set up an environment to perform malware analysis using Ghidra and how to use it in the headless mode. As you progress, you'll use Ghidra scripting to automate the task of identifying vulnerabilities in executable binaries. The book also covers advanced topics such as developing Ghidra plug-ins, developing your own GUI, incorporating new process architectures if needed, and contributing to the Ghidra project. By the end of this Ghidra book, you'll have developed the skills you need to harness the power of Ghidra for analyzing and avoiding potential vulnerabilities in code and networks. What you will learn

Get to grips with using Ghidra's features, plug-ins, and extensions

Understand how you can contribute to Ghidra

Focus on reverse engineering malware and perform binary auditing

Automate reverse engineering tasks with Ghidra plug-ins

Become well-versed with developing your own Ghidra extensions, scripts, and features

Automate the task of looking for vulnerabilities in executable binaries using Ghidra scripting

Find out how to use Ghidra in the headless mode

Who this book is for

This SRE book is for developers, software engineers, or any IT professional with some understanding of cybersecurity essentials. Prior knowledge of Java or Python, along with experience in programming or developing applications, is required before getting started with this book.

A computer forensics "how-to" for fighting malicious code and analyzing incidents

With our ever-increasing reliance on computers comes an ever-growing risk of malware. Security professionals will

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

find plenty of solutions in this book to the problems posed by viruses, Trojan horses, worms, spyware, rootkits, adware, and other invasive software. Written by well-known malware experts, this guide reveals solutions to numerous problems and includes a DVD of custom programs and tools that illustrate the concepts, enhancing your skills. Security professionals face a constant battle against malicious software; this practical manual will improve your analytical capabilities and provide dozens of valuable and innovative solutions. Covers classifying malware, packing and unpacking, dynamic malware analysis, decoding and decrypting, rootkit detection, memory forensics, open source malware research, and much more. Includes generous amounts of source code in C, Python, and Perl to extend your favorite tools or build new ones, and custom programs on the DVD to demonstrate the solutions. Malware Analyst's Cookbook is indispensable to IT security administrators, incident responders, forensic analysts, and malware researchers.

Can we emulate nature's technology in chemistry? Through billions of years of evolution, Nature has generated some remarkable systems and substances that have made life on earth what it is today. Increasingly, scientists are seeking to mimic Nature's systems and processes in the lab in order to harness the power of Nature for the benefit of society. Bioinspiration and Biomimicry in Chemistry explores the chemistry of Nature and how we can replicate what Nature does in abiological settings. Specifically, the book focuses on wholly artificial, man-made systems that employ or are inspired by principles of Nature, but which do not use materials of biological origin. Beginning with a general overview of the concept of bioinspiration and biomimicry in chemistry, the book tackles such topics as: Bioinspired molecular machines, Bioinspired catalysis, Biomimetic amphiphiles and vesicles, Biomimetic principles in macromolecular science, Biomimetic cavities and bioinspired receptors, Biomimicry in organic synthesis. Written by a team of leading international experts, the contributed chapters collectively lay the groundwork for a new generation of environmentally friendly and sustainable materials, pharmaceuticals, and technologies. Readers will discover the latest advances in our ability to replicate natural systems and materials as well as the many impediments that remain, proving how much we still need to learn about how Nature works. Bioinspiration and Biomimicry in Chemistry is recommended for students and researchers in all realms of chemistry. Addressing how scientists are working to reverse engineer Nature in all areas of chemical research, the book is designed to stimulate new discussion and research in this exciting and promising field.

The definitive guide—fully updated for Windows 10 and Windows Server 2016. Delve inside Windows architecture and internals, and see how core components work behind the scenes. Led by a team of internals experts, this classic guide has been fully updated for Windows 10 and Windows Server 2016. Whether you are a developer or an IT professional, you'll get critical, insider perspectives on how Windows operates. And through hands-on experiments, you'll experience its internal behavior firsthand—knowledge you can apply to improve application design, debugging, system performance, and

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

support. This book will help you:

- Understand the Windows system architecture and its most important entities, such as processes and threads
- Examine how processes manage resources and threads scheduled for execution inside processes
- Observe how Windows manages virtual and physical memory
- Dig into the Windows I/O system and see how device drivers work and integrate with the rest of the system
- Go inside the Windows security model to see how it manages access, auditing, and authorization, and learn about the new mechanisms in Windows 10 and Server 2016

Ghidra Software Reverse Engineering for Beginners

The Ghidra Book

The Art of Memory Forensics

Black Hat Go

Modern Computer Architecture and Organization

The Antivirus Hacker's Handbook

Bioinspiration and Biomimicry in Chemistry

Master Shellcode to leverage the buffer overflow concept
Key Features
Understand how systems can be bypassed both at the operating system and network level with shellcode, assembly, and Metasploit
Learn to write and modify 64-bit shellcode along with kernel-level shellcode concepts
A step-by-step guide that will take you from low-level security skills to covering loops with shellcode
Book Description
Security has always been a major concern for your application, your system, or your environment. This book's main goal is to build your skills for low-level security exploits, finding vulnerabilities and covering loopholes with shellcode, assembly, and Metasploit. This book will teach you topics ranging from memory management and assembly to compiling and extracting shellcode and using syscalls and dynamically locating functions in memory. This book also covers techniques to compile 64-bit shellcode for Linux and Windows along with Metasploit shellcode tools. Lastly, this book will also show you to how to write your own exploits with intermediate techniques, using real-world scenarios. By the end of this book, you will have become an expert in shellcode and will understand how systems are compromised both at the operating system and network level. What you will learn
Create an isolated lab to test and inject shellcodes (Windows and Linux). Understand both Windows and Linux behavior. Learn the assembly programming language. Create shellcode using assembly and Metasploit. Detect buffer overflows. Debug and reverse-engineer using tools such as GDB, edb, and Immunity (Windows and Linux). Exploit development and shellcodes injections (Windows & Linux). Prevent and protect against buffer overflows and heap corruption. Who this book is for
This book is intended to be read by penetration testers, malware analysts, security researchers, forensic practitioners, exploit developers, C language programmers, software testers, and students in the security field. Readers should have a basic understanding of OS internals (Windows and Linux). Some knowledge of the C programming language is

Download File PDF Practical Reverse Engineering: X86, X64, ARM, Windows Kernel, Reversing Tools, And Obfuscation

essential, and a familiarity with the Python language would be helpful.

Reverse Engineering

The Real Practice of X86 Internals, Code Calling Conventions, Ransomware Decryption, Application Cracking, Assembly Language, and Proven Cybersecurity Open Source Tools (English Edition)

The Rootkit Arsenal

Reversing

Mastering Malware Analysis

Constructive Engineering of Large Reverse Osmosis Desalination Plants

Build Your Own Linux Tools for Binary Instrumentation, Analysis, and Disassembly