Learning Linux Binary Analysis

Linux ForensicsLinux with Operating System ConceptsUnderstanding the Linux KernelBeginning Ethical Hacking with Kali LinuxAttacking Network ProtocolsThe Art of Memory ForensicsLearning Kali LinuxBuilding Embedded Linux SystemsInformation Theory, Inference and Learning AlgorithmsAdvanced Linux ProgrammingProfessional Linux Kernel ArchitectureKali Linux RevealedARM Assembly Language and Reverse EngineeringMastering Malware AnalysisSELinux CookbookLinux Security CookbookAdvanced C and C++ CompilingPractical Malware AnalysisPractical Reverse EngineeringThe Antivirus Hacker's HandbookBinary Code Fingerprinting for CybersecurityMachine Learning and SecurityPython Machine LearningBinary Analysis CookbookLinkers and LoadersPractical Binary AnalysisHands-On System Programming with LinuxProgramming Embedded SystemsThe Linux Command LineBPF Performance ToolsExtreme CThe Ghidra BookMastering Reverse EngineeringLinux Basics for Hackers40 Algorithms Every Programmer Should KnowLearning Malware AnalysisLinux Device DriversLearning Linux Binary AnalysisReversingHead First Networking

Linux Forensics

Whether you're a veteran or an absolute n00b, this is the best place to start with Kali Linux, the security professional's platform of choice, and a truly industrial-grade, and world-class operating system distribution-

mature, secure, and enterprise-ready.

Linux with Operating System Concepts

You've experienced the shiny, point-and-click surface of your Linux computer—now dive below and explore its depths with the power of the command line. The Linux Command Line takes you from your very first terminal keystrokes to writing full programs in Bash, the most popular Linux shell. Along the way you'll learn the timeless skills handed down by generations of gray-bearded, mouse-shunning gurus: file navigation, environment configuration, command chaining, pattern matching with regular expressions, and more. In addition to that practical knowledge, author William Shotts reveals the philosophy behind these tools and the rich heritage that your desktop Linux machine has inherited from Unix supercomputers of yore. As you make your way through the book's short, easily-digestible chapters, you'll learn how to: * Create and delete files, directories, and symlinks * Administer your system, including networking, package installation, and process management * Use standard input and output, redirection, and pipelines * Edit files with Vi, the world’s most popular text editor * Write shell scripts to automate common or boring tasks * Slice and dice text files with cut, paste, grep, patch, and sed Once you overcome your initial "shell shock," you'll find that the command line is a natural and expressive way to communicate with your computer. Just don't be surprised if your mouse starts to gather dust. A featured resource in the Linux Foundation's

"Evolution of a SysAdmin"

Understanding the Linux Kernel

If you are a Linux system administrator or a Linux-based service administrator and want to fine-tune SELinux to implement a supported, mature, and proven access control system, then this book is for you. Basic experience with SELinux enabled distributions is expected.

Beginning Ethical Hacking with Kali Linux

Attacking Network Protocols is a deep dive into network protocol security from James Forshaw, one of the world's leading bug hunters. This comprehensive guide looks at networking from an attacker's perspective to help you discover, exploit, and ultimately protect vulnerabilities. You'll start with a rundown of networking basics and protocol traffic capture before moving on to static and dynamic protocol analysis, common protocol structures, cryptography, and protocol security. Then you'll turn your focus to finding and exploiting vulnerabilities, with an overview of common bug classes, fuzzing, debugging, and exhaustion attacks. Learn how to: - Capture, manipulate, and replay packets - Develop tools to dissect traffic and reverse engineer code to understand the inner workings of a network protocol - Discover and exploit vulnerabilities such as memory corruptions, authentication bypasses, and denials of service - Use capture and analysis tools like Wireshark and develop your own custom network proxies to

manipulate network traffic Attacking Network Protocols is a must-have for any penetration tester, bug hunter, or developer looking to understand and discover network vulnerabilities.

Attacking Network Protocols

Get up and running with system programming concepts in Linux Key Features Acquire insight on Linux system architecture and its programming interfaces Get to grips with core concepts such as process management, signalling and pthreads Packed with industry best practices and dozens of code examples Book Description The Linux OS and its embedded and server applications are critical components of today’s software infrastructure in a decentralized, networked universe. The industry's demand for proficient Linux developers is only rising with time. Hands-On System Programming with Linux gives you a solid theoretical base and practical industry-relevant descriptions, and covers the Linux system programming domain. It delves into the art and science of Linux application programming—system architecture, process memory and management, signaling, timers, pthreads, and file IO. This book goes beyond the use API X to do Y approach; it explains the concepts and theories required to understand programming interfaces and design decisions, the tradeoffs made by experienced developers when using them, and the rationale behind them. Troubleshooting tips and techniques are included in the concluding chapter. By the end of this book, you will have gained essential conceptual

design knowledge and hands-on experience working with Linux system programming interfaces. What you will learn Explore the theoretical underpinnings of Linux system architecture Understand why modern OSes use virtual memory and dynamic memory APIs Get to grips with dynamic memory issues and effectively debug them Learn key concepts and powerful system APIs related to process management Effectively perform file IO and use signaling and timers Deeply understand multithreading concepts, pthreads APIs, synchronization and scheduling Who this book is for Hands-On System Programming with Linux is for Linux system engineers, programmers, or anyone who wants to go beyond using an API set to understanding the theoretical underpinnings and concepts behind powerful Linux system programming APIs. To get the most out of this book, you should be familiar with Linux at the user-level logging in, using shell via the command line interface, the ability to use tools such as find, grep, and sort. Working knowledge of the C programming language is required. No prior experience with Linux systems programming is assumed.

The Art of Memory Forensics

Analyzing how hacks are done, so as to stop them in thefuture Reverse engineering is the process of analyzing hardware orsoftware and understanding it, without having access to the sourcecode or design documents. Hackers are able to reverse engineersystems and exploit what they find with scary results. Now the goodguys can use the same

tools to thwart these threats. PracticalReverse Engineering goes under the hood of reverse engineeringfor security analysts, security engineers, and system programmers,so they can learn how to use these same processes to stop hackersin their tracks. The book covers x86, x64, and ARM (the first book to cover allthree); Windows kernel-mode code rootkits and drivers; virtualmachine protection techniques; and much more. Best of all, itoffers a systematic approach to the material, with plenty ofhands-on exercises and real-world examples. Offers a systematic approach to understanding reverseengineering, with hands-on exercises and real-world examples Covers x86, x64, and advanced RISC machine (ARM) architecturesas well as deobfuscation and virtual machine protectiontechniques Provides special coverage of Windows kernel-mode code(rootkits/drivers), a topic not often covered elsewhere, andexplains 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, WindowsKernel, and Reversing Tools provides crucial, up-to-dateguidance for a broad range of IT professionals.

Learning Kali Linux

To thoroughly understand what makes Linux tick and why it's so efficient, you need to delve deep into the heart of the operating system--into the Linux kernel itself. The kernel is Linux--in the case of the Linux operating system, it's the only bit of software to which

the term "Linux" applies. The kernel handles all the requests or completed I/O operations and determines which programs will share its processing time, and in what order. Responsible for the sophisticated memory management of the whole system, the Linux kernel is the force behind the legendary Linux efficiency. The new edition of Understanding the Linux Kernel takes you on a guided tour through the most significant data structures, many algorithms, and programming tricks used in the kernel. Probing beyond the superficial features, the authors offer valuable insights to people who want to know how things really work inside their machine. Relevant segments of code are dissected and discussed line by line. The book covers more than just the functioning of the code, it explains the theoretical underpinnings for why Linux does things the way it does. The new edition of the book has been updated to cover version 2.4 of the kernel, which is quite different from version 2.2: the virtual memory system is entirely new, support for multiprocessor systems is improved, and whole new classes of hardware devices have been added. The authors explore each new feature in detail. Other topics in the book include: Memory management including file buffering, process swapping, and Direct memory Access (DMA) The Virtual Filesystem and the Second Extended Filesystem Process creation and scheduling Signals, interrupts, and the essential interfaces to device drivers Timing Synchronization in the kernel Interprocess Communication (IPC) Program execution Understanding the Linux Kernel, Second Edition will acquaint you with all the inner workings of Linux, but is more than just an academic exercise. You'll learn what conditions bring out Linux's best

performance, and you'll see how it meets the challenge of providing good system response during process scheduling, file access, and memory management in a wide variety of environments. If knowledge is power, then this book will help you make the most of your Linux system.

Building Embedded Linux Systems

Learn algorithms for solving classic computer science problems with this concise guide covering everything from fundamental algorithms, such as sorting and searching, to modern algorithms used in machine learning and cryptography Key Features Learn the techniques you need to know to design algorithms for solving complex problems Become familiar with neural networks and deep learning techniques Explore different types of algorithms and choose the right data structures for their optimal implementation Book Description Algorithms have always played an important role in both the science and practice of computing. Beyond traditional computing, the ability to use algorithms to solve real-world problems is an important skill that any developer or programmer must have. This book will help you not only to develop the skills to select and use an algorithm to solve real-world problems but also to understand how it works. You'll start with an introduction to algorithms and discover various algorithm design techniques, before exploring how to implement different types of algorithms, such as searching and sorting, with the help of practical examples. As you advance to a more complex set of algorithms, you'll learn about linear

programming, page ranking, and graphs, and even work with machine learning algorithms, understanding the math and logic behind them. Further on, case studies such as weather prediction, tweet clustering, and movie recommendation engines will show you how to apply these algorithms optimally. Finally, you'll become well versed in techniques that enable parallel processing, giving you the ability to use these algorithms for compute-intensive tasks. By the end of this book, you'll have become adept at solving real-world computational problems by using a wide range of algorithms. What you will learn Explore existing data structures and algorithms found in Python libraries Implement graph algorithms for fraud detection using network analysis Work with machine learning algorithms to cluster similar tweets and process Twitter data in real time Predict the weather using supervised learning algorithms Use neural networks for object detection Create a recommendation engine that suggests relevant movies to subscribers Implement foolproof security using symmetric and asymmetric encryption on Google Cloud Platform (GCP) Who this book is for This book is for the serious programmer! Whether you are an experienced programmer looking to gain a deeper understanding of the math behind the algorithms or have limited programming or data science knowledge and want to learn more about how you can take advantage of these battle-tested algorithms to improve the way you design and write code, you'll find this book useful. Experience with Python programming is a must, although knowledge of data science is helpful but not necessary.

Information Theory, Inference and Learning Algorithms

BPF and related observability tools give software professionals unprecedented visibility into software, helping them analyze operating system and application performance, troubleshoot code, and strengthen security. BPF Performance Tools: Linux System and Application Observability is the industry's most comprehensive guide to using these tools for observability. Brendan Gregg, author of the industry's definitive guide to system performance, introduces powerful new methods and tools for doing analysis that leads to more robust, reliable, and safer code. This authoritative guide: Explores a wide spectrum of software and hardware targets Thoroughly covers open source BPF tools from the Linux Foundation iovisor project's bcc and bpftrace repositories Summarizes performance engineering and kernel internals you need to understand Provides and discusses 150+ bpftrace tools, including 80 written specifically for this book: tools you can run as-is, without programming — or customize and develop further, using diverse interfaces and the bpftrace front-end You'll learn how to use BPF (eBPF) tracing tools to analyze CPUs, memory, disks, file systems, networking, languages, applications, containers, hypervisors, security, and the Linux kernel. You'll move from basic to advanced tools and techniques, producing new metrics, stack traces, custom latency histograms, and more. It's like having a superpower: with Gregg's guidance and tools, you can analyze virtually everything that impacts system

performance, so you can improve virtually any Linux operating system or application.

Advanced Linux Programming

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 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.

Professional Linux Kernel Architecture

Introduces tools and techniques for analyzing and debugging malicious software, discussing how to set up a safe virtual environment, overcome malware tricks, and use five of the most popular packers.

Kali Linux Revealed

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

ARM Assembly Language and Reverse Engineering

Authored by two of the leading authorities in the field, this guide offers readers the knowledge and skills needed to achieve proficiency with embedded software.

Mastering Malware Analysis

Table of contents

SELinux Cookbook

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.

Linux Security Cookbook

Computer security is an ongoing process, a relentless contest between system administrators and intruders. A good administrator needs to stay one step ahead of any adversaries, which often involves a continuing process of education. If you're grounded in the basics of security, however, you won't necessarily want a complete treatise on the subject each time you pick up a book. Sometimes you want to get straight to the point. That's exactly what the new Linux Security Cookbook does. Rather than provide a total security solution for Linux computers, the authors present a series of easy-to-follow recipes--short, focused pieces of code that administrators can use to improve security and perform common tasks securely.The Linux Security Cookbook includes real solutions to a wide range of targeted problems, such as sending encrypted email within Emacs, restricting access to network services at particular times of day, firewalling a webserver, preventing IP spoofing, setting up key-

based SSH authentication, and much more. With over 150 ready-to-use scripts and configuration files, this unique book helps administrators secure their systems without having to look up specific syntax. The book begins with recipes devised to establish a secure system, then moves on to secure day-to-day practices, and concludes with techniques to help your system stay secure.Some of the "recipes" you'll find in this book are: Controlling access to your system from firewalls down to individual services, using iptables, ipchains, xinetd, inetd, and more Monitoring your network with tcpdump, dsniff, netstat, and other tools Protecting network connections with Secure Shell (SSH) and stunnel Safeguarding email sessions with Secure Sockets Layer (SSL) Encrypting files and email messages with GnuPG Probing your own security with password crackers, nmap, and handy scripts This cookbook's proven techniques are derived from hard-won experience. Whether you're responsible for security on a home Linux system or for a large corporation, or somewhere in between, you'll find valuable, to-the-point, practical recipes for dealing with everyday security issues. This book is a system saver.

Advanced C and C++ Compiling

Practical Binary Analysis is the first book of its kind to present advanced binary analysis topics in an accessible way. After an introduction on the basics of binary formats, disassembly, and code injection, you'll dive into more complex topics such as binary instrumentation, dynamic taint analysis, and symbolic

execution. By the end of the book, you'll be able to build your own binary analysis tools on Linux by following hands-on and practical examples.

Practical Malware Analysis

Learning how to write C/C++ code is only the first step. To be a serious programmer, you need to understand the structure and purpose of the binary files produced by the compiler: object files, static libraries, shared libraries, and, of course, executables. Advanced C and C++ Compiling explains the build process in detail and shows how to integrate code from other developers in the form of deployed libraries as well as how to resolve issues and potential mismatches between your own and external code trees. With the proliferation of open source, understanding these issues is increasingly the responsibility of the individual programmer. Advanced C and C++ Compiling brings all of the information needed to move from intermediate to expert programmer together in one place -- an engineering guide on the topic of C/C++ binaries to help you get the most accurate and pertinent information in the quickest possible time.

Practical Reverse Engineering

ARM Assembly Language and Reverse Engineering starts with an introduction to all three flavors of ARM: 32-bit, Thumb, and 64-Bit. It gives readers an introduction into how these operating systems function which will be the basis from which to

investigate, attack, and secure billions of mobile devices, IoT devices, laptops, smartwatches, and more. In covering ARM fundamentals the book covers architecture, privilege levels, and virtual memory. Whatever the reader wants to accomplish with reverse engineering starts with understanding ARM assembly. The book looks at writing assembly, data types and registers, instruction sets and processor modes, instructions, operations, and branches, and how programs are loaded by the OS. Ultimately the book introduces and dives deep into reversing ARM binaries. Readers will explore disassembling ARM binaries on Linux and using the previously secret NSA tool Ghidra to reverse engineer arm binaries. Deep attention is given to dynamic analysis, debugging, flow control patterns, and instruction blocks.

The Antivirus Hacker's Handbook

Find an introduction to the architecture, concepts and algorithms of the Linux kernel in Professional Linux Kernel Architecture, a guide to the kernel sources and large number of connections among subsystems. Find an introduction to the relevant structures and functions exported by the kernel to userland, understand the theoretical and conceptual aspects of the Linux kernel and Unix derivatives, and gain a deeper understanding of the kernel. Learn how to reduce the vast amount of information contained in the kernel sources and obtain the skills necessary to understand the kernel sources.

Binary Code Fingerprinting for

Cybersecurity

This practical, tutorial-style book uses the Kali Linux distribution to teach Linux basics with a focus on how hackers would use them. Topics include Linux command line basics, filesystems, networking, BASH basics, package management, logging, and the Linux kernel and drivers. If you're getting started along the exciting path of hacking, cybersecurity, and pentesting, Linux Basics for Hackers is an excellent first step. Using Kali Linux, an advanced penetration testing distribution of Linux, you'll learn the basics of using the Linux operating system and acquire the tools and techniques you'll need to take control of a Linux environment. First, you'll learn how to install Kali on a virtual machine and get an introduction to basic Linux concepts. Next, you'll tackle broader Linux topics like manipulating text, controlling file and directory permissions, and managing user environment variables. You'll then focus in on foundational hacking concepts like security and anonymity and learn scripting skills with bash and Python. Practical tutorials and exercises throughout will reinforce and test your skills as you learn how to: - Cover your tracks by changing your network information and manipulating the rsyslog logging utility - Write a tool to scan for network connections, and connect and listen to wireless networks - Keep your internet activity stealthy using Tor, proxy servers, VPNs, and encrypted email - Write a bash script to scan open ports for potential targets - Use and abuse services like MySQL, Apache web server, and OpenSSH - Build your own hacking tools, such as

a remote video spy camera and a password cracker Hacking is complex, and there is no single way in. Why not start at the beginning with Linux Basics for Hackers?

Machine Learning and Security

Unlock deeper insights into Machine Leaning with this vital guide to cutting-edge predictive analytics About This Book Leverage Python's most powerful open-source libraries for deep learning, data wrangling, and data visualization Learn effective strategies and best practices to improve and optimize machine learning systems and algorithms Ask – and answer – tough questions of your data with robust statistical models, built for a range of datasets Who This Book Is For If you want to find out how to use Python to start answering critical questions of your data, pick up Python Machine Learning – whether you want to get started from scratch or want to extend your data science knowledge, this is an essential and unmissable resource. What You Will Learn Explore how to use different machine learning models to ask different questions of your data Learn how to build neural networks using Keras and Theano Find out how to write clean and elegant Python code that will optimize the strength of your algorithms Discover how to embed your machine learning model in a web application for increased accessibility Predict continuous target outcomes using regression analysis Uncover hidden patterns and structures in data with clustering Organize data using effective pre-processing techniques Get to grips with sentiment

analysis to delve deeper into textual and social media data In Detail Machine learning and predictive analytics are transforming the way businesses and other organizations operate. Being able to understand trends and patterns in complex data is critical to success, becoming one of the key strategies for unlocking growth in a challenging contemporary marketplace. Python can help you deliver key insights into your data – its unique capabilities as a language let you build sophisticated algorithms and statistical models that can reveal new perspectives and answer key questions that are vital for success. Python Machine Learning gives you access to the world of predictive analytics and demonstrates why Python is one of the world's leading data science languages. If you want to ask better questions of data, or need to improve and extend the capabilities of your machine learning systems, this practical data science book is invaluable. Covering a wide range of powerful Python libraries, including scikit-learn, Theano, and Keras, and featuring guidance and tips on everything from sentiment analysis to neural networks, you'll soon be able to answer some of the most important questions facing you and your organization. Style and approach Python Machine Learning connects the fundamental theoretical principles behind machine learning to their practical application in a way that focuses you on asking and answering the right questions. It walks you through the key elements of Python and its powerful machine learning libraries, while demonstrating how to get to grips with a range of statistical models.

Python Machine Learning

With more than 600 security tools in its arsenal, the Kali Linux distribution can be overwhelming. Experienced and aspiring security professionals alike may find it challenging to select the most appropriate tool for conducting a given test. This practical book covers Kali's expansive security capabilities and helps you identify the tools you need to conduct a wide range of security tests and penetration tests. You'll also explore the vulnerabilities that make those tests necessary. Author Ric Messier takes you through the foundations of Kali Linux and explains methods for conducting tests on networks, web applications, wireless security, password vulnerability, and more. You'll discover different techniques for extending Kali tools and creating your own toolset. Learn tools for stress testing network stacks and applications Perform network reconnaissance to determine what's available to attackers Execute penetration tests using automated exploit tools such as Metasploit Use cracking tools to see if passwords meet complexity requirements Test wireless capabilities by injecting frames and cracking passwords Assess web application vulnerabilities with automated or proxy-based tools Create advanced attack techniques by extending Kali tools or developing your own Use Kali Linux to generate reports once testing is complete

Binary Analysis Cookbook

Provides information on writing a driver in Linux, covering such topics as character devices, network interfaces, driver debugging, concurrency, and interrupts.

Linkers and Loaders

This book addresses automated software fingerprinting in binary code, especially for cybersecurity applications. The reader will gain a thorough understanding of binary code analysis and several software fingerprinting techniques for cybersecurity applications, such as malware detection, vulnerability analysis, and digital forensics. More specifically, it starts with an overview of binary code analysis and its challenges, and then discusses the existing state-of-the-art approaches and their cybersecurity applications. Furthermore, it discusses and details a set of practical techniques for compiler provenance extraction, library function identification, function fingerprinting, code reuse detection, free open-source software identification, vulnerability search, and authorship attribution. It also illustrates several case studies to demonstrate the efficiency, scalability and accuracy of the above-mentioned proposed techniques and tools. This book also introduces several innovative quantitative and qualitative techniques that synergistically leverage machine learning, program analysis, and software engineering methods to solve binary code fingerprinting problems, which are highly relevant to cybersecurity and digital forensics applications. The above-mentioned techniques are cautiously designed to gain satisfactory levels of efficiency and accuracy. Researchers working in academia, industry and governmental agencies focusing on Cybersecurity will want to purchase this book. Software engineers and advanced-level students studying computer science,

computer engineering and software engineering will also want to purchase this book.

Practical Binary Analysis

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 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.

Hands-On System Programming with Linux

Get started in white-hat ethical hacking using Kali Linux. This book starts off by giving you an overview of security trends, where you will learn the OSI security architecture. This will form the foundation for the rest of Beginning Ethical Hacking with Kali Linux. With the theory out of the way, you'll move on to an introduction to VirtualBox, networking, and common Linux commands, followed by the step-by-step procedure to build your own web server and acquire the skill to be anonymous . When you have finished the examples in the first part of your book, you will have all you need to carry out safe and ethical hacking experiments. After an introduction to Kali Linux, you will carry out your first penetration tests with Python and code raw binary packets for use in those tests. You will learn how to find secret directories on a target system, use a TCP client in Python, and scan ports using NMAP. Along the way you will discover effective ways to collect important information, track email, and use important tools such as DMITRY and Maltego, as well as take a look at the five phases of penetration testing. The coverage of vulnerability analysis includes sniffing and spoofing, why ARP poisoning is a threat, how SniffJoke prevents poisoning, how to analyze protocols with Wireshark,

and using sniffing packets with Scapy. The next part of the book shows you detecting SQL injection vulnerabilities, using sqlmap, and applying brute force or password attacks. Besides learning these tools, you will see how to use OpenVas, Nikto, Vega, and Burp Suite. The book will explain the information assurance model and the hacking framework Metasploit, taking you through important commands, exploit and payload basics. Moving on to hashes and passwords you will learn password testing and hacking techniques with John the Ripper and Rainbow. You will then dive into classic and modern encryption techniques where you will learn the conventional cryptosystem. In the final chapter you will acquire the skill of exploiting remote Windows and Linux systems and you will learn how to own a target completely. What You Will Learn Master common Linux commands and networking techniques Build your own Kali web server and learn to be anonymous Carry out penetration testing using Python Detect sniffing attacks and SQL injection vulnerabilities Learn tools such as SniffJoke, Wireshark, Scapy, sqlmap, OpenVas, Nikto, and Burp Suite Use Metasploit with Kali Linux Exploit remote Windows and Linux systems Who This Book Is For Developers new to ethical hacking with a basic understanding of Linux programming.

Programming Embedded Systems

Can machine learning techniques solve our computer security problems and finally put an end to the cat-and-mouse game between attackers and defenders?

Or is this hope merely hype? Now you can dive into the science and answer this question for yourself! With this practical guide, you'll explore ways to apply machine learning to security issues such as intrusion detection, malware classification, and network analysis. Machine learning and security specialists Clarence Chio and David Freeman provide a framework for discussing the marriage of these two fields, as well as a toolkit of machine-learning algorithms that you can apply to an array of security problems. This book is ideal for security engineers and data scientists alike. Learn how machine learning has contributed to the success of modern spam filters Quickly detect anomalies, including breaches, fraud, and impending system failure Conduct malware analysis by extracting useful information from computer binaries Uncover attackers within the network by finding patterns inside datasets Examine how attackers exploit consumer-facing websites and app functionality Translate your machine learning algorithms from the lab to production Understand the threat attackers pose to machine learning solutions

The Linux Command Line

This is the eBook version of the printed book. If the print book includes a CD-ROM, this content is not included within the eBook version. Advanced Linux Programming is divided into two parts. The first covers generic UNIX system services, but with a particular eye towards Linux specific information. This portion of the book will be of use even to advanced programmers who have worked with other Linux

systems since it will cover Linux specific details and differences. For programmers without UNIX experience, it will be even more valuable. The second section covers material that is entirely Linux specific. These are truly advanced topics, and are the techniques that the gurus use to build great applications. While this book will focus mostly on the Application Programming Interface (API) provided by the Linux kernel and the C library, a preliminary introduction to the development tools available will allow all who purchase the book to make immediate use of Linux.

BPF Performance Tools

Linux Forensics is the most comprehensive and up-to-date resource for those wishing to quickly and efficiently perform forensicson Linux systems. It is also a great asset for anyone that would like to better understand Linux internals. Linux Forensics will guide you step by step through the process of investigating a computer running Linux. Everything you need to know from the moment you receive the call from someone who thinks they have been attacked until the final report is written is covered in this book. All of the tools discussed in this book are free and most are also open source. Dr. Philip Polstra shows how to leverage numerous tools such as Python, shell scripting, and MySQL to quickly, easily, and accurately analyze Linux systems. While readers will have a strong grasp of Python and shell scripting by the time they complete this book, no priorknowledge of either of these scripting languages is assumed.

Linux Forensics begins by showing you how to determine if there was an incident with minimally invasive techniques. Once it appears likely that an incident has occurred, Dr. Polstra shows you how to collect data from a live system before shutting it down for the creation of filesystem images. Linux Forensics contains extensive coverage of Linux ext2, ext3, and ext4 filesystems. A large collection of Python and shell scripts for creating, mounting, and analyzing filesystem images are presented in this book. Dr. Polstra introduces readers to the exciting new field of memory analysis using the Volatility framework. Discussions of advanced attacks and malware analysis round out the book. Book Highlights 370 pages in large, easy-to-read 8.5 x 11 inch format Over 9000 lines of Python scripts with explanations Over 800 lines of shell scripts with explanations A 102 page chapter containing up-to-date information on the ext4 filesystem Two scenarios described in detail with images available from the book website All scripts and other support files are available from the book website Chapter Contents First Steps General Principles Phases of Investigation High-level Process Building a Toolkit Determining If There Was an Incident Opening a Case Talking to Users Documenation Mounting Known-good Binaries Minimizing Disturbance to the Subject Automation With Scripting Live Analysis Getting Metadata Using Spreadsheets Getting Command Histories Getting Logs Using Hashes Dumping RAM Creating Images Shutting Down the System Image Formats DD DCFLDD Write Blocking Imaging Virtual Machines Imaging Physical Drives Mounting Images Master Boot Record Based Partions GUID Partition Tables Mounting

Partitions In Linux Automating With Python Analyzing Mounted Images Getting Timestamps Using LibreOffice Using MySQL Creating Timelines Extended Filesystems Basics Superblocks Features Using Python Finding Things That Are Out Of Place Inodes Journaling Memory Analysis Volatility Creating Profiles Linux Commands Dealing With More Advanced Attackers Malware Is It Malware? Malware Analysis Tools Static Analysis Dynamic Analysis Obfuscation The Road Ahead Learning More Communities Conferences Certifications

Extreme C

A True Textbook for an Introductory Course, System Administration Course, or a Combination Course Linux with Operating System Concepts merges conceptual operating system (OS) and Unix/Linux topics into one cohesive textbook for undergraduate students. The book can be used for a one- or two-semester course on Linux or Unix. It is complete with review sections, problems, definitions, concepts, and relevant introductory material, such as binary and Boolean logic, OS kernels, and the role of the CPU and memory hierarchy. Details for Introductory and Advanced Users The book covers Linux from both the user and system administrator positions. From a user perspective, it emphasizes command line interaction. From a system administrator perspective, the text reinforces shell scripting with examples of administration scripts that support the automation of administrator tasks. Thorough Coverage of Concepts and Linux Commands The author incorporates OS

concepts not found in most Linux/Unix textbooks, including kernels, file systems, storage devices, virtual memory, and process management. He also introduces computer science topics, such as computer networks and TCP/IP, binary numbers and Boolean logic, encryption, and the GNUs C compiler. In addition, the text discusses disaster recovery planning, booting, and Internet servers.

The Ghidra Book

Linux® is being adopted by an increasing number of embedded systems developers, who have been won over by its sophisticated scheduling and networking, its cost-free license, its open development model, and the support offered by rich and powerful programming tools. While there is a great deal of hype surrounding the use of Linux in embedded systems, there is not a lot of practical information. Building Embedded Linux Systems is the first in-depth, hard-core guide to putting together an embedded system based on the Linux kernel. This indispensable book features arcane and previously undocumented procedures for: Building your own GNU development toolchain Using an efficient embedded development framework Selecting, configuring, building, and installing a target-specific kernel Creating a complete target root filesystem Setting up, manipulating, and using solid-state storage devices Installing and configuring a bootloader for the target Cross-compiling a slew of utilities and packages Debugging your embedded system using a plethora of tools and techniques

Details are provided for various target architectures and hardware configurations, including a thorough review of Linux's support for embedded hardware. All explanations rely on the use of open source and free software packages. By presenting how to build the operating system components from pristine sources and how to find more documentation or help, this book greatly simplifies the task of keeping complete control over one's embedded operating system, whether it be for technical or sound financial reasons.Author Karim Yaghmour, a well-known designer and speaker who is responsible for the Linux Trace Toolkit, starts by discussing the strengths and weaknesses of Linux as an embedded operating system. Licensing issues are included, followed by a discussion of the basics of building embedded Linux systems. The configuration, setup, and use of over forty different open source and free software packages commonly used in embedded Linux systems are also covered. uClibc, BusyBox, U-Boot, OpenSSH, thttpd, tftp, strace, and gdb are among the packages discussed.

Mastering Reverse Engineering

"I enjoyed reading this useful overview of the techniques and challenges of implementing linkers and loaders. While most of the examples are focused on three computer architectures that are widely used today, there are also many side comments about interesting and quirky computer architectures of the past. I can tell from these war stories that the author really has been there himself and survived to tell the

tale." -Guy Steele Whatever your programming language, whatever your platform, you probably tap into linker and loader functions all the time. But do you know how to use them to their greatest possible advantage? Only now, with the publication of Linkers & Loaders, is there an authoritative book devoted entirely to these deep-seated compile-time and run-time processes. The book begins with a detailed and comparative account of linking and loading that illustrates the differences among various compilers and operating systems. On top of this foundation, the author presents clear practical advice to help you create faster, cleaner code. You'll learn to avoid the pitfalls associated with Windows DLLs, take advantage of the space-saving, performance-improving techniques supported by many modern linkers, make the best use of the UNIX ELF library scheme, and much more. If you're serious about programming, you'll devour this unique guide to one of the field's least understood topics. Linkers & Loaders is also an ideal supplementary text for compiler and operating systems courses. Features: * Includes a linker construction project written in Perl, with project files available for download. * Covers dynamic linking in Windows, UNIX, Linux, BeOS, and other operating systems. * Explains the Java linking model and how it figures in network applets and extensible Java code. * Helps you write more elegant and effective code, and build applications that compile, load, and run more efficiently.

Linux Basics for Hackers

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 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.

40 Algorithms Every Programmer Should Know

Frustrated with networking books so chock-full of acronyms that your brain goes into sleep mode? Head First Networking's unique, visually rich format provides a task-based approach to computer networking that makes it easy to get your brain engaged. You'll learn the concepts by tying them to on-the-job tasks, blending practice and theory in a way that only Head First can. With this book, you'll learn skills through a variety of genuine scenarios, from fixing a malfunctioning office network to planning a network for a high-technology haunted house. You'll learn exactly what you need to know, rather than a laundry list of acronyms and diagrams.

This book will help you: Master the functionality, protocols, and packets that make up real-world networking Learn networking concepts through examples in the field Tackle tasks such as planning and diagramming networks, running cables, and configuring network devices such as routers and switches Monitor networks for performance and problems, and learn troubleshooting techniques Practice what you've learned with nearly one hundred exercises, questions, sample problems, and projects Head First's popular format is proven to stimulate learning and retention by engaging you with images, puzzles, stories, and more. Whether you're a network professional with a CCNA/CCNP or a student taking your first college networking course, Head First Networking will help you become a network guru.

Learning Malware Analysis

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

Linux Device Drivers

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.

Learning Linux Binary Analysis

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.

Reversing

Push the limits of what C - and you - can do, with this high-intensity guide to the most advanced capabilities of C Key Features Make the most of C's low-level control, flexibility, and high performance A comprehensive guide to C's most powerful and challenging features A thought-provoking guide packed with hands-on exercises and examples Book Description There's a lot more to C than knowing the language syntax. The industry looks for developers with a rigorous, scientific understanding of the principles and practices. Extreme C will teach you to use C's advanced low-level power to write effective, efficient systems. This intensive, practical guide will help you become an expert C programmer. Building on your existing C knowledge, you will master preprocessor directives, macros, conditional compilation, pointers, and much more. You will gain new insight into algorithm design, functions, and structures. You will discover how C helps you squeeze maximum performance out of critical, resource-constrained applications. C still plays a critical role in 21st-century programming, remaining the core language for precision engineering, aviations, space research, and more. This book shows how C works with Unix, how to implement OO principles in C, and fully covers multi-processing. In Extreme C, Amini encourages you to think, question, apply, and experiment for yourself. The book is essential for anybody who wants to take their C to the next level. What you will learn Build advanced C knowledge on strong foundations, rooted in first principles Understand memory structures and compilation pipeline and how they work, and how to make most out of them Apply object-oriented design principles to

your procedural C code Write low-level code that's close to the hardware and squeezes maximum performance out of a computer system Master concurrency, multithreading, multi-processing, and integration with other languages Unit Testing and debugging, build systems, and inter-process communication for C programming Who this book is for Extreme C is for C programmers who want to dig deep into the language and its capabilities. It will help you make the most of the low-level control C gives you.

Head First Networking

Uncover the secrets of Linux binary analysis with this handy guide About This Book Grasp the intricacies of the ELF binary format of UNIX and Linux Design tools for reverse engineering and binary forensic analysis Insights into UNIX and Linux memory infections, ELF viruses, and binary protection schemes Who This Book Is For If you are a software engineer or reverse engineer and want to learn more about Linux binary analysis, this book will provide you with all you need to implement solutions for binary analysis in areas of security, forensics, and antivirus. This book is great for both security enthusiasts and system level engineers. Some experience with the C programming language and the Linux command line is assumed. What You Will Learn Explore the internal workings of the ELF binary format Discover techniques for UNIX Virus infection and analysis Work with binary hardening and software anti-tamper methods Patch executables and process memory Bypass anti-

debugging measures used in malware Perform advanced forensic analysis of binaries Design ELF-related tools in the C language Learn to operate on memory with ptrace In Detail Learning Linux Binary Analysis is packed with knowledge and code that will teach you the inner workings of the ELF format, and the methods used by hackers and security analysts for virus analysis, binary patching, software protection and more. This book will start by taking you through UNIX/Linux object utilities, and will move on to teaching you all about the ELF specimen. You will learn about process tracing, and will explore the different types of Linux and UNIX viruses, and how you can make use of ELF Virus Technology to deal with them. The latter half of the book discusses the usage of Kprobe instrumentation for kernel hacking, code patching, and debugging. You will discover how to detect and disinfect kernel-mode rootkits, and move on to analyze static code. Finally, you will be walked through complex userspace memory infection analysis. This book will lead you into territory that is uncharted even by some experts; right into the world of the computer hacker. Style and approach The material in this book provides detailed insight into the arcane arts of hacking, coding, reverse engineering Linux executables, and dissecting process memory. In the computer security industry these skills are priceless, and scarce. The tutorials are filled with knowledge gained through first hand experience, and are complemented with frequent examples including source code.