

Writing Secure Code 2nd Edition Developer Best Practices

This is likewise one of the factors by obtaining the soft documents of this **Writing Secure Code 2nd Edition Developer Best Practices** by online. You might not require more become old to spend to go to the book introduction as competently as search for them. In some cases, you likewise pull off not discover the message Writing Secure Code 2nd Edition Developer Best Practices that you are looking for. It will totally squander the time.

However below, similar to you visit this web page, it will be thus completely simple to get as capably as download guide Writing Secure Code 2nd Edition Developer Best Practices

It will not say yes many period as we accustom before. You can complete it though take steps something else at house and even in your workplace. for that reason easy! So, are you question? Just exercise just what we present below as with ease as evaluation **Writing Secure Code 2nd Edition Developer Best Practices** what you taking into consideration to read!

The Hitchhiker's Guide to Python - Kenneth Reitz 2016-08-30

The Hitchhiker's Guide to Python takes the journeyman Pythonista to true expertise. More than any other language, Python was created with the philosophy of simplicity and parsimony. Now 25 years old, Python has become the primary or secondary language (after SQL) for many business users. With popularity comes diversity—and possibly dilution. This guide, collaboratively written by over a hundred members of the Python community, describes best practices currently used by package and application developers. Unlike other books for this audience, The Hitchhiker's Guide is light on reusable code and heavier on design philosophy, directing the reader to excellent sources that already exist.

Writing Secure Code - David LeBlanc 2002-12-04

Keep black-hat hackers at bay with the tips and techniques in this entertaining, eye-opening book! Developers will learn how to padlock their applications throughout the entire development process—from designing secure applications to writing robust code that can withstand repeated attacks to testing applications for security flaws. Easily digested chapters reveal proven principles, strategies, and coding techniques. The authors—two battle-scarred veterans who

have solved some of the industry's toughest security problems—provide sample code in several languages. This edition includes updated information about threat modeling, designing a security process, international issues, file-system issues, adding privacy to applications, and performing security code reviews. It also includes enhanced coverage of buffer overruns, Microsoft .NET security, and Microsoft ActiveX development, plus practical checklists for developers, testers, and program managers.

[24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them](#) - Michael Howard 2009-09-22

"What makes this book so important is that it reflects the experiences of two of the industry's most experienced hands at getting real-world engineers to understand just what they're being asked for when they're asked to write secure code. The book reflects Michael Howard's and David LeBlanc's experience in the trenches working with developers years after code was long since shipped, informing them of problems." --From the Foreword by Dan Kaminsky, Director of Penetration Testing, IOActive Eradicate the Most Notorious Insecure Designs and Coding Vulnerabilities Fully updated to cover the latest security issues, 24 Deadly Sins of Software Security reveals the most common design and coding errors and

explains how to fix each one-or better yet, avoid them from the start. Michael Howard and David LeBlanc, who teach Microsoft employees and the world how to secure code, have partnered again with John Viega, who uncovered the original 19 deadly programming sins. They have completely revised the book to address the most recent vulnerabilities and have added five brand-new sins. This practical guide covers all platforms, languages, and types of applications. Eliminate these security flaws from your code: SQL injection Web server- and client-related vulnerabilities Use of magic URLs, predictable cookies, and hidden form fields Buffer overruns Format string problems Integer overflows C++ catastrophes Insecure exception handling Command injection Failure to handle errors Information leakage Race conditions Poor usability Not updating easily Executing code with too much privilege Failure to protect stored data Insecure mobile code Use of weak password-based systems Weak random numbers Using cryptography incorrectly Failing to protect network traffic Improper use of PKI Trusting network name resolution

Alice and Bob Learn Application Security - Tanya Janca 2020-10-09

Learn application security from the very start, with this comprehensive and approachable guide! Alice and Bob Learn Application Security is an accessible and thorough resource for anyone seeking to incorporate, from the beginning of the System Development Life Cycle, best security practices in software development. This book covers all the basic subjects such as threat modeling and security testing, but also dives deep into more complex and advanced topics for securing modern software systems and architectures. Throughout, the book offers analogies, stories of the characters Alice and Bob, real-life examples, technical explanations and diagrams to ensure maximum clarity of the many abstract and complicated subjects. Topics include: Secure requirements, design, coding, and deployment Security Testing (all forms) Common Pitfalls Application Security Programs Securing Modern Applications Software Developer Security Hygiene Alice and Bob Learn Application Security is perfect for aspiring application security engineers and practicing software

developers, as well as software project managers, penetration testers, and chief information security officers who seek to build or improve their application security programs. Alice and Bob Learn Application Security illustrates all the included concepts with easy-to-understand examples and concrete practical applications, furthering the reader's ability to grasp and retain the foundational and advanced topics contained within.

Secure Programming with Static Analysis - Brian Chess 2007-06-29

The First Expert Guide to Static Analysis for Software Security! Creating secure code requires more than just good intentions.

Programmers need to know that their code will be safe in an almost infinite number of scenarios and configurations. Static source code analysis gives users the ability to review their work with a fine-toothed comb and uncover the kinds of errors that lead directly to security

vulnerabilities. Now, there's a complete guide to static analysis: how it works, how to integrate it into the software development processes, and how to make the most of it during security code review. Static analysis experts Brian Chess and Jacob West look at the most common types of security defects that occur today. They illustrate main points using Java and C code examples taken from real-world security incidents, showing how coding errors are exploited, how they could have been prevented, and how static analysis can rapidly uncover similar mistakes.

This book is for everyone concerned with building more secure software: developers, security engineers, analysts, and testers.

Secure Coding in C and C++ - Robert C. Seacord 2005-09-09

"The security of information systems has not improved at a rate consistent with the growth and sophistication of the attacks being made against them. To address this problem, we must improve the underlying strategies and techniques used to create our systems.

Specifically, we must build security in from the start, rather than append it as an afterthought. That's the point of Secure Coding in C and C++. In careful detail, this book shows software developers how to build high-quality systems that are less vulnerable to costly and even catastrophic attack. It's a book that every

developer should read before the start of any serious project." --Frank Abagnale, author, lecturer, and leading consultant on fraud prevention and secure documents Learn the Root Causes of Software Vulnerabilities and How to Avoid Them Commonly exploited software vulnerabilities are usually caused by avoidable software defects. Having analyzed nearly 18,000 vulnerability reports over the past ten years, the CERT/Coordination Center (CERT/CC) has determined that a relatively small number of root causes account for most of them. This book identifies and explains these causes and shows the steps that can be taken to prevent exploitation. Moreover, this book encourages programmers to adopt security best practices and develop a security mindset that can help protect software from tomorrow's attacks, not just today's. Drawing on the CERT/CC's reports and conclusions, Robert Seacord systematically identifies the program errors most likely to lead to security breaches, shows how they can be exploited, reviews the potential consequences, and presents secure alternatives. Coverage includes technical detail on how to Improve the overall security of any C/C++ application Thwart buffer overflows and stack-smashing attacks that exploit insecure string manipulation logic Avoid vulnerabilities and security flaws resulting from the incorrect use of dynamic memory management functions Eliminate integer-related problems: integer overflows, sign errors, and truncation errors Correctly use formatted output functions without introducing format-string vulnerabilities Avoid I/O vulnerabilities, including race conditions Secure Coding in C and C++ presents hundreds of examples of secure code, insecure code, and exploits, implemented for Windows and Linux. If you're responsible for creating secure C or C++ software--or for keeping it safe--no other book offers you this much detailed, expert assistance.

Linux System Programming - Robert Love
2013-05-14

UNIX, UNIX LINUX & UNIX TCL/TK. Write software that makes the most effective use of the Linux system, including the kernel and core system libraries. The majority of both Unix and Linux code is still written at the system level, and this book helps you focus on everything above the kernel, where applications such as

Apache, bash, cp, vim, Emacs, gcc, gdb, glibc, ls, mv, and X exist. Written primarily for engineers looking to program at the low level, this updated edition of Linux System Programming gives you an understanding of core internals that makes for better code, no matter where it appears in the stack. -- Provided by publisher.

Hacking the Code - Mark Burnett 2004-05-10
Hacking the Code has over 400 pages of dedicated exploit, vulnerability, and tool code with corresponding instruction. Unlike other security and programming books that dedicate hundreds of pages to architecture and theory based flaws and exploits, Hacking the Code dives right into deep code analysis. Previously undisclosed security research in combination with superior programming techniques from Foundstone and other respected organizations is included in both the Local and Remote Code sections of the book. The book is accompanied with a FREE COMPANION CD containing both commented and uncommented versions of the source code examples presented throughout the book. In addition to the book source code, the CD also contains a copy of the author-developed Hacker Code Library v1.0. The Hacker Code Library includes multiple attack classes and functions that can be utilized to quickly create security programs and scripts. These classes and functions simplify exploit and vulnerability tool development to an extent never before possible with publicly available software. Learn to quickly create security tools that ease the burden of software testing and network administration Find out about key security issues regarding vulnerabilities, exploits, programming flaws, and secure code development Discover the differences in numerous types of web-based attacks so that developers can create proper quality assurance testing procedures and tools Learn to automate quality assurance, management, and development tasks and procedures for testing systems and applications Learn to write complex Snort rules based solely upon traffic generated by network tools and exploits

Full Stack Python Security - Dennis Byrne
2021-08-24

Full Stack Python Security teaches you everything you'll need to build secure Python web applications. Summary In Full Stack Python

Security: Cryptography, TLS, and attack resistance, you'll learn how to: Use algorithms to encrypt, hash, and digitally sign data Create and install TLS certificates Implement authentication, authorization, OAuth 2.0, and form validation in Django Protect a web application with Content Security Policy Implement Cross Origin Resource Sharing Protect against common attacks including clickjacking, denial of service attacks, SQL injection, cross-site scripting, and more Full Stack Python Security: Cryptography, TLS, and attack resistance teaches you everything you'll need to build secure Python web applications. As you work through the insightful code snippets and engaging examples, you'll put security standards, best practices, and more into action. Along the way, you'll get exposure to important libraries and tools in the Python ecosystem. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Security is a full-stack concern, encompassing user interfaces, APIs, web servers, network infrastructure, and everything in between. Master the powerful libraries, frameworks, and tools in the Python ecosystem and you can protect your systems top to bottom. Packed with realistic examples, lucid illustrations, and working code, this book shows you exactly how to secure Python-based web applications. About the book Full Stack Python Security: Cryptography, TLS, and attack resistance teaches you everything you need to secure Python and Django-based web apps. In it, seasoned security pro Dennis Byrne demystifies complex security terms and algorithms. Starting with a clear review of cryptographic foundations, you'll learn how to implement layers of defense, secure user authentication and third-party access, and protect your applications against common hacks. What's inside Encrypt, hash, and digitally sign data Create and install TLS certificates Implement authentication, authorization, OAuth 2.0, and form validation in Django Protect against attacks such as clickjacking, cross-site scripting, and SQL injection About the reader For intermediate Python programmers. About the author Dennis Byrne is a tech lead for 23andMe, where he protects the genetic data of more than 10 million

customers. Table of Contents 1 Defense in depth PART 1 - CRYPTOGRAPHIC FOUNDATIONS 2 Hashing 3 Keyed hashing 4 Symmetric encryption 5 Asymmetric encryption 6 Transport Layer Security PART 2 - AUTHENTICATION AND AUTHORIZATION 7 HTTP session management 8 User authentication 9 User password management 10 Authorization 11 OAuth 2 PART 3 - ATTACK RESISTANCE 12 Working with the operating system 13 Never trust input 14 Cross-site scripting attacks 15 Content Security Policy 16 Cross-site request forgery 17 Cross-Origin Resource Sharing 18 Clickjacking *Designing Secure Software* - Loren Kohnfelder 2021-12-21 What every software professional should know about security. *Designing Secure Software* consolidates Loren Kohnfelder's more than twenty years of experience into a concise, elegant guide to improving the security of technology products. Written for a wide range of software professionals, it emphasizes building security into software design early and involving the entire team in the process. The book begins with a discussion of core concepts like trust, threats, mitigation, secure design patterns, and cryptography. The second part, perhaps this book's most unique and important contribution to the field, covers the process of designing and reviewing a software design with security considerations in mind. The final section details the most common coding flaws that create vulnerabilities, making copious use of code snippets written in C and Python to illustrate implementation vulnerabilities. You'll learn how to:

- Identify important assets, the attack surface, and the trust boundaries in a system
- Evaluate the effectiveness of various threat mitigation candidates
- Work with well-known secure coding patterns and libraries
- Understand and prevent vulnerabilities like XSS and CSRF, memory flaws, and more
- Use security testing to proactively identify vulnerabilities introduced into code
- Review a software design for security flaws effectively and without judgment

Kohnfelder's career, spanning decades at Microsoft and Google, introduced numerous software security initiatives, including the co-creation of the STRIDE threat modeling framework used widely today. This book is a

modern, pragmatic consolidation of his best practices, insights, and ideas about the future of software.

Building Secure Software - John Viega
2001-09-24

Most organizations have a firewall, antivirus software, and intrusion detection systems, all of which are intended to keep attackers out. So why is computer security a bigger problem today than ever before? The answer is simple--bad software lies at the heart of all computer security problems. Traditional solutions simply treat the symptoms, not the problem, and usually do so in a reactive way. This book teaches you how to take a proactive approach to computer security. *Building Secure Software* cuts to the heart of computer security to help you get security right the first time. If you are serious about computer security, you need to read this book, which includes essential lessons for both security professionals who have come to realize that software is the problem, and software developers who intend to make their code behave. Written for anyone involved in software development and use—from managers to coders—this book is your first step toward building more secure software. *Building Secure Software* provides expert perspectives and techniques to help you ensure the security of essential software. If you consider threats and vulnerabilities early in the development cycle you can build security into your system. With this book you will learn how to determine an acceptable level of risk, develop security tests, and plug security holes before software is even shipped. Inside you'll find the ten guiding principles for software security, as well as detailed coverage of: Software risk management for security Selecting technologies to make your code more secure Security implications of open source and proprietary software How to audit software The dreaded buffer overflow Access control and password authentication Random number generation Applying cryptography Trust management and input Client-side security Dealing with firewalls Only by building secure software can you defend yourself against security breaches and gain the confidence that comes with knowing you won't have to play the "penetrate and patch" game anymore. Get it right the first time. Let these expert authors

show you how to properly design your system; save time, money, and credibility; and preserve your customers' trust.

Code Craft - Pete Goodliffe 2007

A guide to writing computer code covers such topics as variable naming, presentation style, error handling, and security.

The Security Development Lifecycle - Michael Howard 2006

Describes how to put software security into practice, covering such topics as risk analysis, coding policies, Agile Methods, cryptographic standards, and threat tree patterns.

Code Complete - Steve McConnell 2004-06-09

Widely considered one of the best practical guides to programming, Steve McConnell's original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. No matter what your experience level, development environment, or project size, this book will inform and stimulate your thinking—and help you build the highest quality code. Discover the timeless techniques and strategies that help you: Design for minimum complexity and maximum creativity Reap the benefits of collaborative development Apply defensive programming techniques to reduce and flush out errors Exploit opportunities to refactor—or evolve—code, and do it safely Use construction practices that are right-weight for your project Debug problems quickly and effectively Resolve critical construction issues early and correctly Build quality into the beginning, middle, and end of your project

Effective C - Robert C. Seacord 2020-08-11

A detailed introduction to the C programming language for experienced programmers. The world runs on code written in the C programming language, yet most schools begin the curriculum with Python or Java. *Effective C* bridges this gap and brings C into the modern era--covering the modern C17 Standard as well

as potential C2x features. With the aid of this instant classic, you'll soon be writing professional, portable, and secure C programs to power robust systems and solve real-world problems. Robert C. Seacord introduces C and the C Standard Library while addressing best practices, common errors, and open debates in the C community. Developed together with other C Standards committee experts, *Effective C* will teach you how to debug, test, and analyze C programs. You'll benefit from Seacord's concise explanations of C language constructs and behaviors, and from his 40 years of coding experience. You'll learn:

- How to identify and handle undefined behavior in a C program
- The range and representations of integers and floating-point values
- How dynamic memory allocation works and how to use nonstandard functions
- How to use character encodings and types
- How to perform I/O with terminals and filesystems using C Standard streams and POSIX file descriptors
- How to understand the C compiler's translation phases and the role of the preprocessor
- How to test, debug, and analyze C programs

Effective C will teach you how to write professional, secure, and portable C code that will stand the test of time and help strengthen the foundation of the computing world.

Foundations of Security - Christoph Kern
2007-05-11

Software developers need to worry about security as never before. They need clear guidance on safe coding practices, and that's exactly what this book delivers. The book does not delve deep into theory, or rant about the politics of security. Instead, it clearly and simply lays out the most common threats that programmers need to defend against. It then shows programmers how to make their defense. The book takes a broad focus, ranging over SQL injection, worms and buffer overflows, password security, and more. It sets programmers on the path towards successfully defending against the entire gamut of security threats that they might face.

Programming Embedded Systems - Michael Barr
2006-10-11

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.

Beautiful Code - Greg Wilson 2007-06-26
How do the experts solve difficult problems in software development? In this unique and insightful book, leading computer scientists offer case studies that reveal how they found unusual, carefully designed solutions to high-profile projects. You will be able to look over the shoulder of major coding and design experts to see problems through their eyes. This is not simply another design patterns book, or another software engineering treatise on the right and wrong way to do things. The authors think aloud as they work through their project's architecture, the tradeoffs made in its construction, and when it was important to break rules. This book contains 33 chapters contributed by Brian Kernighan, Karl Fogel, Jon Bentley, Tim Bray, Elliotte Rusty Harold, Michael Feathers, Alberto Savoia, Charles Petzold, Douglas Crockford, Henry S. Warren, Jr., Ashish Gulhati, Lincoln Stein, Jim Kent, Jack Dongarra and Piotr Luszczek, Adam Kolawa, Greg Kroah-Hartman, Diomidis Spinellis, Andrew Kuchling, Travis E. Oliphant, Ronald Mak, Rogerio Atem de Carvalho and Rafael Monnerat, Bryan Cantrill, Jeff Dean and Sanjay Ghemawat, Simon Peyton Jones, Kent Dybvig, William Otte and Douglas C. Schmidt, Andrew Patzer, Andreas Zeller, Yukihiro Matsumoto, Arun Mehta, TV Raman, Laura Wingerd and Christopher Seiwald, and Brian Hayes. *Beautiful Code* is an opportunity for master coders to tell their story. All author royalties will be donated to Amnesty International.

Write Better, Faster - Monica Leonelle
2020-12-15

In 2012, fiction author Monica Leonelle made a life-changing decision to learn to write faster. Through months of trial-and-error, hundreds of hours of experimentation, and dozens of manuscripts, she tweaked and honed until she could easily write 10,000 words in a day, at speeds over 3500+ words per hour! She shares all her insights, secrets, hacks, and data in this tome dedicated to improving your writing speeds, skyrocketing your monthly word count, and publishing more books. You'll learn:

- The Writing Faster Framework that Monica used to reach speeds of 3500+ new fiction words per

hour - The tracking systems you need to double or triple your writing speed in the next couple months - The killer 4-step pre-production method Monica uses to combat writer's block, no matter what the project is! - The secrets to developing a daily writing habit that other authors don't talk about enough - How Monica went from publishing only one book per year from 2009-2013, to publishing 8 books in a single year in 2014 For serious authors, both beginner and advanced, who want to improve their output this year! Write Better, Faster: How To Triple Your Writing Speed and Write More Every Day will help you kick your excuses and get more writing done. As part of The Productive Novelist series, it explores how to hack your writing routine to be more efficient, more productive, and have a ton of fun in the process! *Security for Web Developers* - John Paul Mueller 2015-11-10

As a web developer, you may not want to spend time making your web app secure, but it definitely comes with the territory. This practical guide provides you with the latest information on how to thwart security threats at several levels, including new areas such as microservices. You'll learn how to help protect your app no matter where it runs, from the latest smartphone to an older desktop, and everything in between. Author John Paul Mueller delivers specific advice as well as several security programming examples for developers with a good knowledge of CSS3, HTML5, and JavaScript. In five separate sections, this book shows you how to protect against viruses, DDoS attacks, security breaches, and other nasty intrusions. Create a security plan for your organization that takes the latest devices and user needs into account Develop secure interfaces, and safely incorporate third-party code from libraries, APIs, and microservices Use sandboxing techniques, in-house and third-party testing techniques, and learn to think like a hacker Implement a maintenance cycle by determining when and how to update your application software Learn techniques for efficiently tracking security threats as well as training requirements that your organization can use

Write Great Code, Volume 2, 2nd Edition - Randall Hyde 2020-08-11

Thinking Low-Level, Writing High-Level, the second volume in the landmark Write Great Code series by Randall Hyde, covers high-level programming languages (such as Swift and Java) as well as code generation on 64-bit CPUs ARM, the Java Virtual Machine, and the Microsoft Common Runtime. Today's programming languages offer productivity and portability, but also make it easy to write sloppy code that isn't optimized for a compiler. Thinking Low-Level, Writing High-Level will teach you to craft source code that results in good machine code once it's run through a compiler. You'll learn: How to analyze the output of a compiler to verify that your code generates good machine code The types of machine code statements that compilers generate for common control structures, so you can choose the best statements when writing HLL code Enough assembly language to read compiler output How compilers convert various constant and variable objects into machine data With an understanding of how compilers work, you'll be able to write source code that they can translate into elegant machine code. NEW TO THIS EDITION, COVERAGE OF: Programming languages like Swift and Java Code generation on modern 64-bit CPUs ARM processors on mobile phones and tablets Stack-based architectures like the Java Virtual Machine Modern language systems like the Microsoft Common Language Runtime

Writing Secure Code for Windows Vista - Michael Howard 2007

Provides information on writing more secure code for Microsoft Windows Vista, covering such topics as application compatibility, buffer overrun defenses, network security, Windows CardSpace, parental controls, and Windows Defender APIs.

Good Code, Bad Code - Tom Long 2021-09-07

"For coders early in their careers who are familiar with an object-oriented language, such as Java or C#"--Back cover.

The Pragmatic Programmer - Andrew Hunt 1999-10-20

What others in the trenches say about The Pragmatic Programmer... "The cool thing about this book is that it's great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there." —Kent Beck,

author of *Extreme Programming Explained: Embrace Change* "I found this book to be a great mix of solid advice and wonderful analogies!" —Martin Fowler, author of *Refactoring* and *UML Distilled* "I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost." —Kevin Ruland, Management Science, MSG-Logistics "The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful.... By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and expert mentors alike." —John Lakos, author of *Large-Scale C++ Software Design* "This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients." —Eric Vought, Software Engineer "Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book." —Pete McBreen, Independent Consultant "Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my company time and money while helping me get my job done quicker! This should be a desktop reference for everyone who works with code for a living." —Jared Richardson, Senior Software Developer, iRenaissance, Inc. "I would like to see this issued to every new employee at my company...." —Chris Cleeland, Senior Software Engineer, Object Computing, Inc. "If I'm putting together a project, it's the authors of this book that I want. . . . And failing that I'd settle for people who've read their book." —Ward Cunningham Straight from the programming trenches, *The Pragmatic Programmer* cuts through the increasing specialization and technicalities of modern software development to examine the core process--taking a

requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting analogies, *The Pragmatic Programmer* illustrates the best practices and major pitfalls of many different aspects of software development. Whether you're a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you'll quickly see improvements in personal productivity, accuracy, and job satisfaction. You'll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You'll become a Pragmatic Programmer.

Secure by Design - Daniel Sawano 2019-09-03
As a developer, you need to build software in a secure way. But you can't spend all your time focusing on security. The answer is to use good design principles, tools, and mindsets that make security an implicit result - it's secure by design. *Secure by Design* teaches developers how to use design to drive security in software development. This book is full of patterns, best practices, and mindsets that you can directly apply to your real world development. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

[ASP.NET Core 5 Secure Coding Cookbook](#) - Roman Canlas 2021-07-16

Learn how to secure your ASP.NET Core web app through robust and secure code Key FeaturesDiscover the different types of security weaknesses in ASP.NET Core web applications and learn how to fix themUnderstand what code

makes an ASP.NET Core web app unsafeBuild your secure coding knowledge by following straightforward recipesBook Description ASP.NET Core developers are often presented with security test results showing the vulnerabilities found in their web apps. While the report may provide some high-level fix suggestions, it does not specify the exact steps that you need to take to resolve or fix weaknesses discovered by these tests. In ASP.NET Secure Coding Cookbook, you'll start by learning the fundamental concepts of secure coding and then gradually progress to identifying common web app vulnerabilities in code. As you progress, you'll cover recipes for fixing security misconfigurations in ASP.NET Core web apps. The book further demonstrates how you can resolve different types of Cross-Site Scripting. A dedicated section also takes you through fixing miscellaneous vulnerabilities that are no longer in the OWASP Top 10 list. This book features a recipe-style format, with each recipe containing sample insecure code that presents the problem and corresponding solutions to eliminate the security bug. You'll be able to follow along with each step of the exercise and use the accompanying sample ASP.NET Core solution to practice writing secure code. By the end of this book, you'll be able to identify insecure code causing different security flaws in ASP.NET Core web apps and you'll have gained hands-on experience in removing vulnerabilities and security defects from your code. What you will learnUnderstand techniques for squashing an ASP.NET Core web app security bugDiscover different types of injection attacks and understand how you can prevent this vulnerability from being exploitedFix security issues in code relating to broken authentication and authorizationEliminate the risks of sensitive data exposure by getting up to speed with numerous protection techniquesPrevent security misconfiguration by enabling ASP.NET Core web application security featuresExplore other ASP.NET web application vulnerabilities and secure coding best practicesWho this book is for This ASP.NET Core book is for intermediate-level ASP.NET Core web developers and software engineers who use the framework to develop web applications and are looking to

focus on their security using coding best practices. The book is also for application security engineers, analysts, and specialists who want to know more about securing ASP.NET Core using code and understand how to resolve issues identified by the security tests they perform daily.

The Rust Programming Language (Covers Rust 2018) - Steve Klabnik 2019-09-03

The official book on the Rust programming language, written by the Rust development team at the Mozilla Foundation, fully updated for Rust 2018. The Rust Programming Language is the official book on Rust: an open source systems programming language that helps you write faster, more reliable software. Rust offers control over low-level details (such as memory usage) in combination with high-level ergonomics, eliminating the hassle traditionally associated with low-level languages. The authors of The Rust Programming Language, members of the Rust Core Team, share their knowledge and experience to show you how to take full advantage of Rust's features--from installation to creating robust and scalable programs. You'll begin with basics like creating functions, choosing data types, and binding variables and then move on to more advanced concepts, such as:

- Ownership and borrowing, lifetimes, and traits
- Using Rust's memory safety guarantees to build fast, safe programs
- Testing, error handling, and effective refactoring
- Generics, smart pointers, multithreading, trait objects, and advanced pattern matching
- Using Cargo, Rust's built-in package manager, to build, test, and document your code and manage dependencies
- How best to use Rust's advanced compiler with compiler-led programming techniques

You'll find plenty of code examples throughout the book, as well as three chapters dedicated to building complete projects to test your learning: a number guessing game, a Rust implementation of a command line tool, and a multithreaded server. New to this edition: An extended section on Rust macros, an expanded chapter on modules, and appendixes on Rust development tools and editions.

Write Great Code, Volume 1, 2nd Edition - Randall Hyde 2020-07-31

Understanding the Machine, the first volume in the landmark Write Great Code series by Randall

Hyde, explains the underlying mechanics of how a computer works. This, the first volume in Randall Hyde's Write Great Code series, dives into machine organization without the extra overhead of learning assembly language programming. Written for high-level language programmers, Understanding the Machine fills in the low-level details of machine organization that are often left out of computer science and engineering courses. Learn: How the machine represents numbers, strings, and high-level data structures, so you'll know the inherent cost of using them. How to organize your data, so the machine can access it efficiently. How the CPU operates, so you can write code that works the way the machine does. How I/O devices operate, so you can maximize your application's performance when accessing those devices. How to best use the memory hierarchy to produce the fastest possible programs. Great code is efficient code. But before you can write truly efficient code, you must understand how computer systems execute programs and how abstractions in programming languages map to the machine's low-level hardware. After all, compilers don't write the best machine code; programmers do. This book gives you the foundation upon which all great software is built. NEW IN THIS EDITION, COVERAGE OF: Programming languages like Swift and Java Code generation on modern 64-bit CPUs ARM processors on mobile phones and tablets Newer peripheral devices Larger memory systems and large-scale SSDs

Secure Coding - Mark G. Graff 2003-06

Despite their myriad manifestations and different targets, nearly all attacks on computer systems have one fundamental cause: the code used to run far too many systems today is not secure. Flaws in its design, implementation, testing, and operations allow attackers all-too-easy access. "Secure Coding, by Mark G. Graff and Ken vanWyk, looks at the problem of bad code in a new way. Packed with advice based on the authors' decades of experience in the computer security field, this concise and highly readable book explains why so much code today is filled with vulnerabilities, and tells readers what they must do to avoid writing code that can be exploited by attackers. Beyond the technical, "Secure Coding sheds new light on the

economic, psychological, and sheer practical reasons why security vulnerabilities are so ubiquitous today. It presents a new way of thinking about these vulnerabilities and ways that developers can compensate for the factors that have produced such unsecured software in the past. It issues a challenge to all those concerned about computer security to finally make a commitment to building code the right way.

Framework Design Guidelines - Krzysztof Cwalina 2008-10-22

This is the eBook version of the print title, Framework Design Guidelines, Second Edition . Access to all the samples, applications, and content on the DVD is available through the product catalog page

www.informit.com/title/9780321545619

Navigate to the "Downloads" tab and click on the "DVD Contents" links - see instructions in back pages of your eBook. Framework Design Guidelines, Second Edition, teaches developers the best practices for designing reusable libraries for the Microsoft .NET Framework. Expanded and updated for .NET 3.5, this new edition focuses on the design issues that directly affect the programmability of a class library, specifically its publicly accessible APIs. This book can improve the work of any .NET developer producing code that other developers will use. It includes copious annotations to the guidelines by thirty-five prominent architects and practitioners of the .NET Framework, providing a lively discussion of the reasons for the guidelines as well as examples of when to break those guidelines. Microsoft architects Krzysztof Cwalina and Brad Abrams teach framework design from the top down. From their significant combined experience and deep insight, you will learn The general philosophy and fundamental principles of framework design Naming guidelines for the various parts of a framework Guidelines for the design and extending of types and members of types Issues affecting-and guidelines for ensuring-extensibility How (and how not) to design exceptions Guidelines for-and examples of-common framework design patterns Guidelines in this book are presented in four major forms: Do, Consider, Avoid, and Do not. These directives help focus attention on

practices that should always be used, those that should generally be used, those that should rarely be used, and those that should never be used. Every guideline includes a discussion of its applicability, and most include a code example to help illuminate the dialogue. Framework Design Guidelines, Second Edition, is the only definitive source of best practices for managed code API development, direct from the architects themselves. A companion DVD includes the Designing .NET Class Libraries video series, instructional presentations by the authors on design guidelines for developing classes and components that extend the .NET Framework. A sample API specification and other useful resources and tools are also included.

24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them - John Viega 2009-09-24

"What makes this book so important is that it reflects the experiences of two of the industry's most experienced hands at getting real-world engineers to understand just what they're being asked for when they're asked to write secure code. The book reflects Michael Howard's and David LeBlanc's experience in the trenches working with developers years after code was long since shipped, informing them of problems." --From the Foreword by Dan Kaminsky, Director of Penetration Testing, IOActive Eradicate the Most Notorious Insecure Designs and Coding Vulnerabilities Fully updated to cover the latest security issues, 24 Deadly Sins of Software Security reveals the most common design and coding errors and explains how to fix each one-or better yet, avoid them from the start. Michael Howard and David LeBlanc, who teach Microsoft employees and the world how to secure code, have partnered again with John Viega, who uncovered the original 19 deadly programming sins. They have completely revised the book to address the most recent vulnerabilities and have added five brand-new sins. This practical guide covers all platforms, languages, and types of applications. Eliminate these security flaws from your code: SQL injection Web server- and client-related vulnerabilities Use of magic URLs, predictable cookies, and hidden form fields Buffer overruns Format string problems Integer overflows C++ catastrophes Insecure exception handling

Command injection Failure to handle errors Information leakage Race conditions Poor usability Not updating easily Executing code with too much privilege Failure to protect stored data Insecure mobile code Use of weak password-based systems Weak random numbers Using cryptography incorrectly Failing to protect network traffic Improper use of PKI Trusting network name resolution

Secure Programming Cookbook for C and C++ - John Viega 2003-07-14

Password sniffing, spoofing, buffer overflows, and denial of service: these are only a few of the attacks on today's computer systems and networks. At the root of this epidemic is poorly written, poorly tested, and insecure code that puts everyone at risk. Clearly, today's developers need help figuring out how to write code that attackers won't be able to exploit. But writing such code is surprisingly difficult. Secure Programming Cookbook for C and C++ is an important new resource for developers serious about writing secure code. It contains a wealth of solutions to problems faced by those who care about the security of their applications. It covers a wide range of topics, including safe initialization, access control, input validation, symmetric and public key cryptography, cryptographic hashes and MACs, authentication and key exchange, PKI, random numbers, and anti-tampering. The rich set of code samples provided in the book's more than 200 recipes will help programmers secure the C and C++ programs they write for both Unix® (including Linux®) and Windows® environments. Readers will learn: How to avoid common programming errors, such as buffer overflows, race conditions, and format string problems How to properly SSL-enable applications How to create secure channels for client-server communication without SSL How to integrate Public Key Infrastructure (PKI) into applications Best practices for using cryptography properly Techniques and strategies for properly validating input to programs How to launch programs securely How to use file access mechanisms properly Techniques for protecting applications from reverse engineering The book's web site supplements the book by providing a place to post new recipes, including those written in additional languages like Perl,

Java, and Python. Monthly prizes will reward the best recipes submitted by readers. *Secure Programming Cookbook for C and C++* is destined to become an essential part of any developer's library, a code companion developers will turn to again and again as they seek to protect their systems from attackers and reduce the risks they face in today's dangerous world.

[Programming Erlang](#) - Joe Armstrong
2013-09-23

A multi-user game, web site, cloud application, or networked database can have thousands of users all interacting at the same time. You need a powerful, industrial-strength tool to handle the really hard problems inherent in parallel, concurrent environments. You need Erlang. In this second edition of the bestselling *Programming Erlang*, you'll learn how to write parallel programs that scale effortlessly on multicore systems. Using Erlang, you'll be surprised at how easy it becomes to deal with parallel problems, and how much faster and more efficiently your programs run. That's because Erlang uses sets of parallel processes—not a single sequential process, as found in most programming languages. Joe Armstrong, creator of Erlang, introduces this powerful language in small steps, giving you a complete overview of Erlang and how to use it in common scenarios. You'll start with sequential programming, move to parallel programming and handling errors in parallel programs, and learn to work confidently with distributed programming and the standard Erlang/Open Telecom Platform (OTP) frameworks. You need no previous knowledge of functional or parallel programming. The chapters are packed with hands-on, real-world tutorial examples and insider tips and advice, and finish with exercises for both beginning and advanced users. The second edition has been extensively rewritten. New to this edition are seven chapters covering the latest Erlang features: maps, the type system and the Dialyzer, WebSockets, programming idioms, and a new stand-alone execution environment. You'll write programs that dynamically detect and correct errors, and that can be upgraded without stopping the system. There's also coverage of rebar (the de facto Erlang build system), and information on how to share and use Erlang

projects on github, illustrated with examples from cowboy and bitcask. Erlang will change your view of the world, and of how you program. What You Need The Erlang/OTP system.

Download it from erlang.org.

ASP.NET Core Security - Christian Wenz
2022-08-16

Secure your ASP.NET applications before you get hacked! This practical guide includes secure coding techniques with annotated examples and full coverage of built-in ASP.NET Core security tools. In *ASP.NET Core Security*, you will learn how to: Understand and recognize common web app attacks Implement attack countermeasures Use testing and scanning tools and libraries Activate built-in browser security features from ASP.NET Take advantage of .NET and ASP.NET Core security APIs Manage passwords to minimize damage from a data leak Securely store application secrets ASP.NET Core Security teaches you the skills and countermeasures you need to keep your ASP.NET Core apps secure from the most common web application attacks. With this collection of practical techniques, you will be able to anticipate risks and introduce practices like testing as regular security checkups. You'll be fascinated as the author explores real-world security breaches, including rogue Firefox extensions and Adobe password thefts. The examples present universal security best practices with a sharp focus on the unique needs of ASP.NET Core applications. About the technology Your ASP.NET Core applications are under attack now. Are you ready? There are specific countermeasures you can apply to keep your company out of the headlines. This book demonstrates exactly how to secure ASP.NET Core web applications, including safe browser interactions, recognizing common threats, and deploying the framework's unique security APIs. About the book *ASP.NET Core Security* is a realistic guide to securing your web applications. It starts on the dark side, exploring case studies of cross-site scripting, SQL injection, and other weapons used by hackers. As you go, you'll learn how to implement countermeasures, activate browser security features, minimize attack damage, and securely store application secrets. Detailed ASP.NET Core code samples in C# show you how each technique looks in practice. What's inside

Understand and recognize common web app attacks Testing tools, helper libraries, and scanning tools Activate built-in browser security features Take advantage of .NET and ASP.NET Core security APIs Manage passwords to minimize damage from a data leak About the reader For experienced ASP.NET Core web developers. About the author Christian Wenz is a web pioneer, consultant, and entrepreneur.

Table of Contents PART 1 FIRST STEPS 1 On web application security PART 2 MITIGATING COMMON ATTACKS 2 Cross-site scripting (XSS) 3 Attacking session management 4 Cross-site request forgery 5 Unvalidated data 6 SQL injection (and other injections) PART 3 SECURE DATA STORAGE 7 Storing secrets 8 Handling passwords PART 4 CONFIGURATION 9 HTTP headers 10 Error handling 11 Logging and health checks PART 5 AUTHENTICATION AND AUTHORIZATION 12 Securing web applications with ASP.NET Core Identity 13 Securing APIs and single page applications PART 6 SECURITY AS A PROCESS 14 Secure dependencies 15 Audit tools 16 OWASP Top 10

Building Secure and Reliable Systems - Heather Adkins 2020-03-16

Can a system be considered truly reliable if it isn't fundamentally secure? Or can it be considered secure if it's unreliable? Security is crucial to the design and operation of scalable systems in production, as it plays an important part in product quality, performance, and availability. In this book, experts from Google share best practices to help your organization design scalable and reliable systems that are fundamentally secure. Two previous O'Reilly books from Google—*Site Reliability Engineering* and *The Site Reliability*

Workbook—demonstrated how and why a commitment to the entire service lifecycle enables organizations to successfully build, deploy, monitor, and maintain software systems. In this latest guide, the authors offer insights into system design, implementation, and maintenance from practitioners who specialize in security and reliability. They also discuss how building and adopting their recommended best practices requires a culture that's supportive of such change. You'll learn about secure and reliable systems through: Design strategies Recommendations for coding, testing, and

debugging practices Strategies to prepare for, respond to, and recover from incidents Cultural best practices that help teams across your organization collaborate effectively

Writing Secure Code - Michael Howard 2003 Covers topics such as the importance of secure systems, threat modeling, canonical representation issues, solving database input, denial-of-service attacks, and security code reviews and checklists.

Visual Studio 2019 Tricks and Techniques - Paul Schroeder 2021-01-15

Harness the full power of the Visual Studio IDE to take your coding skills to the next level by learning about IDE productivity practices and exclusive techniques Key Features Increase your productivity by leveraging Visual Studio 2019's improvements and features Explore powerful editing, code intelligence, and source code control features to increase productivity Delve into VS's powerful, untapped features such as custom project templates and extensions Book Description Visual Studio 2019 (VS 2019) and Visual Studio Code (VS Code) are powerful professional development tools that help you to develop applications for any platform with ease. Whether you want to create web, mobile, or desktop applications, Microsoft Visual Studio is your one-stop solution. This book demonstrates some of the most sophisticated capabilities of the tooling and shows you how to use the integrated development environment (IDE) more efficiently to be more productive. You'll begin by gradually building on concepts, starting with the basics. The introductory chapters cover shortcuts, snippets, and numerous optimization tricks, along with debugging techniques, source control integration, and other important IDE features that will help you make your time more productive. With that groundwork in place, more advanced concepts such as the inner workings of project and item templates are covered. You will also learn how to write quality, secure code more efficiently as well as discover how certain Visual Studio features work 'under the hood'. By the end of this Visual Studio book, you'll have learned how to write more secure code faster than ever using your knowledge of the extensions and processes that make developing successful solutions more enjoyable and repeatable. What you will learn Understand the

similarities and differences between VS 2019 and VS CodeGet to grips with numerous keyboard shortcuts to improve efficiencyDiscover IDE tips and tricks that make it easier to write codeExperiment with code snippets that make it easier to write repeating code patternsFind out how to customize project and item templates with the help of hands-on exercisesUse Visual Studio extensions for ease and improved productivityDelve into Visual Studio's behind the scene operationsWho this book is for This book is for C# and .NET developers who want to become more efficient and take advantage of features they may not be aware of in the IDE. Those looking to increase their productivity and write quality code more quickly by fully utilizing the power of the Visual Studio IDE will also find this book useful.

Refactoring - Paul Becker 1999

Refactoring is gaining momentum amongst the object oriented programming community. It can transform the internal dynamics of applications and has the capacity to transform bad code into good code. This book offers an introduction to refactoring.

Clean Code - Robert C. Martin 2009

Looks at the principles and clean code, includes case studies showcasing the practices of writing clean code, and contains a list of heuristics and "smells" accumulated from the process of writing clean code.

Seriously Good Software - Marco Faella
2020-03-24

Summary Serious developers know that code can always be improved. With each iteration, you make optimizations—small and large—that can have a huge impact on your application's speed, size, resilience, and maintainability. In *Seriously Good Software: Code that Works, Survives, and Wins*, author, teacher, and Java expert Marco Faella teaches you techniques for writing better code. You'll start with a simple application and follow it through seven careful refactorings, each designed to explore another

dimension of quality. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Great code blends the skill of a programmer with the time-tested techniques and best practices embraced by the entire development community. Although each application has its own context and character, some dimensions of quality are always important. This book concentrates on eight pillars of seriously good software: speed, memory usage, reliability, readability, thread safety, generality, and elegance. The Java-based examples demonstrate techniques that apply to any OO language. About the book *Seriously Good Software* is a handbook for any professional developer serious about improving application quality. It explores fundamental dimensions of code quality by enhancing a simple implementation into a robust, professional-quality application. Questions, exercises, and Java-based examples ensure you'll get a firm grasp of the concepts as you go. When you finish the last version of the book's central project, you'll be able to confidently choose the right optimizations for your code. What's inside Evaluating software qualities Assessing trade-offs and interactions Fulfilling different objectives in a single task Java-based exercises you can apply in any OO language About the reader For web developers comfortable with JavaScript and HTML. About the author Marco Faella teaches advanced programming at a major Italian university. His published work includes peer-reviewed research articles, a Java certification manual, and a video course. Table of Contents *Part 1: Preliminaries * 1 Software qualities and a problem to solve 2 Reference implementation *Part 2: Software Qualities* 3 Need for speed: Time efficiency 4 Precious memory: Space efficiency 5 Self-conscious code: Reliability through monitoring 6 Lie to me: Reliability through testing 7 Coding aloud: Readability 8 Many cooks in the kitchen: Thread safety 9 Please recycle: Reusability