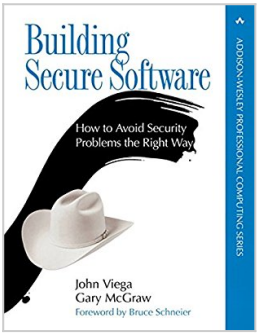

Building Secure Software: How to Avoid Security Problems the Right Way

By John Viega, Gary McGraw

Building Secure Software: How to Avoid Security Problems the Right Way
By John Viega, Gary McGraw

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 devel-opment 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.

Building Secure Software: How to Avoid Security Problems the Right Way

By John Viega, Gary McGraw

Building Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw

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 devel-opment 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.

Building Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw Bibliography

- Sales Rank: #1180363 in Books
- Published on: 2001-10-04
- Original language: English
- Number of items: 1
- Dimensions: 9.38" h x 1.26" w x 7.56" l, 2.16 pounds
- Binding: Hardcover
- 528 pages

Editorial Review

From the Back Cover

"This book is useful, practical, understandable, and comprehensive. The fact that you have this book in your hands is a step in the right direction. Read it, learn from it. And then put its lessons into practice." --From the Foreword by Bruce Schneier, CTO, Counterpane, and author of *Secrets and Lies* "A must-read for anyone writing software for the Internet." --Jeremy Epstein, Director, Product Security and Performance, webMethods "This book tackles complex application security problems like buffer overflows, race conditions, and applied cryptography in a manner that is straightforward and easy to understand. This is a must for any application developer or security professional." --Paul Raines, Global Head of Information Risk Management, Barclays Capital

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 devel-opment 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:

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.

020172152XB08202001 About the Author
John Viega is the CTO of Secure Software Solutions (www.securesw.com) and a noted expert in the area of software security. He is responsible for numerous tools in this area, including code scanners (ITS4 and RATS), random number suites (EGADS), automated repair tools, and secure programming libraries. He is also the original author of Mailman, the GNU mailing list manager. **Gary McGraw,** Cigital's CTO, is a leading authority on software security. Dr. McGraw is coauthor of the groundbreaking books *Building Secure Software* and *Exploiting Software* (both from Addison-Wesley). While consulting for major software producers and consumers, he has published over ninety peer-reviewed technical publications, and functions as principal investigator on grants from DARPA, the National Science Foundation, and NIST's Advanced

Technology Program. He serves on the advisory boards of Authentica, Counterpane, and Fortify Software. He is also an advisor to the computer science departments at University of California, Davis, and the University of Virginia, as well as the School of Informatics at Indiana University.

"A book is a machine to think with."
--I.A. Richards *PRINCIPLES OF LITERARY CRITICISM* This book exists to help people involved in the software development process learn the principles necessary for building secure software. The book is intended for *anyone* involved in software development, from managers to coders, although it contains the low-level detail that is most applicable to programmers. Specific code examples and technical details are presented in the second part of the book. The first part is more general and is intended to set an appropriate context for building secure software by introducing security goals, security technologies, and the concept of software risk management.There are plenty of technical books that deal with computer security, but until now, none have applied significant effort to the topic of developing secure programs. If you want to learn how to set up a firewall, lock down a single host, or build a virtual private network, there are other resources to which to turn outside this book. Because most security books are intended to address the pressing concerns of network-level security practitioners, they tend to focus on how to promote secrecy and how to protect networked resources in a world in which software is chronically broken.Unfortunately, many security practitioners have gotten used to a world in which having security problems in software is common, and even acceptable. Some people even assume that it is too hard to get developers to build secure software, so they don't raise the issue. Instead, they focus their efforts on "best-practice" network security solutions, erecting firewalls, and trying to detect intrusions and patch known security problems in a timely manner.We are optimistic that the problem of bad software security can be addressed. The truth is, writing programs that have no security flaws in them *is* difficult. However, we assert that writing a "secure-enough" program is much easier than writing a completely bug-free program. Should people give up on removing bugs from software just because it's essentially impossible to eliminate them all? Of course not. By the same token, people shouldn't just automatically throw in the software security towel before they even understand the problem.A little bit of education can go a long way. One of the biggest reasons why so many products have security problems is that many technologists involved in the development process have never learned very much about how to produce secure code. One problem is that until now there have been very few places to turn for good information. A goal of this book is to close the educational gap and to arm software practitioners with the basic techniques necessary to write secure programs.This said, you should not expect to eradicate all security problems in your software simply by reading this book. Claiming that this book provides a silver bullet for security would ignore the realities of how difficult it is to secure computer software. We don't ignore reality--we embrace it, by treating software security as a risk management problem.In the real world, your software will likely never be totally secure. First of all, there is no such thing as 100% security. Most software has security risks that can be exploited. It's a matter of how much money and effort are required to break the system in question. Even if your software is bug free and your servers are protected by firewalls, someone who wants to target you may get an insider to attack you. Or they may perform a "black bag" (break-in) operation. Because security is complicated and is a system-wide property, we not only provide general principles for secure software design, but we also focus on the most common risks, and how to mitigate them.OrganizationThis book is divided into two parts. The first part focuses on the things you should know about software security before you even think about producing code. We focus on how to integrate security into your software engineering practice. Emphasis is placed on methodologies and principles that reduce security risk by getting started early in the development life cycle. Designing security into a system from the beginning is much easier and orders of magnitude cheaper than retrofitting a system for security later. Not only do we focus on requirements and design, we also provide significant emphasis on analyzing the security of a system, which we believe to be a critical skill. The first part of this book should be of general interest to anyone involved in software development at any level, from business-level

leadership to developers in the trenches.In the second part, we get our hands dirty with implementation-level issues. Even with a solid architecture, there is plenty of room for security problems to be introduced at development time. We show developers in gory detail how to recognize and to avoid common implementation-level problems such as buffer overflows and race conditions. The second part of the book is intended for those who feel comfortable around code.We purposely cover material that we believe to be of general applicability. That is, unless a topic is security critical, we try to stay away from anything that is dependent on a particular operating system or programming language. For example, we do not discuss POSIX "capabilities" because they are not widely implemented. However, we devote an entire chapter to buffer overflows because they are a problem of extraordinary magnitude, even though a majority of buffer overflows are specific to C and C++.Because our focus is on technologies that are applicable at the broadest levels, there are plenty of worthy technologies that we do not cover, including Kerberos, PAM (pluggable authentication modules), and mobile code sandboxing, to name a few. Many of these technologies merit their own books (although not all of them are adequately covered today). This book's companion Web site, http://www.buildingsecuresoftware.com/, provides links to information sources covering interesting security technologies that we left out.Code ExamplesAlthough we cover material that is largely language independent, most of our examples are written in C, mainly because it is so widely used, but also because it is harder to get things right in C than in other languages. Porting our example code to other programming languages is often a matter of finding the right calls or constructs for the target programming language. However, we do include occasional code examples in Python, Java, and Perl, generally in situations in which those languages are significantly different from C. All of the code in this book is available at http://www.buildingsecuresoftware.com/.There is a large UNIX bias to this book even though we tried to stick to operating system-independent principles. We admit that our coverage of specifics for other operating systems, particularly Windows, leaves something to be desired. Although Windows NT is loosely POSIX compliant, in reality Windows programmers tend not to use the POSIX application programming interface (API). For instance, we hear that most Windows programmers do not use the standard C string library, in favor of Unicode string-handling routines. As of this writing, we still don't know which common functions in the Windows API are susceptible to buffer overflow calls, so we can't provide a comprehensive list. If someone creates such a list in the future, we will gladly post it on the book's Web site.The code we provide in this book has all been tested on a machine running stock Red Hat 6.2. Most of it has been tested on an OpenBSD machine as well. However, we provide the code on an "as-is" basis. We try to make sure that the versions of the code posted on the Web site are as portable as possible; but be forewarned, our available resources for ensuring portability are low. We may not have time to help people who can't get code to compile on a particular architecture, but we will be very receptive to readers who send in patches.Contacting UsWe welcome electronic mail from anyone with comments, bug fixes, or other suggestions. Please contact us through http://www.buildingsecuresoftware.com.

020172152XP09242001 Users Review**From reader reviews:**

Inez Tuller:The book Building Secure Software: How to Avoid Security Problems the Right Way give you a sense of feeling enjoy for your spare time. You can utilize to make your capable considerably more increase. Book can for being your best friend when you getting tension or having big problem with your subject. If you can make reading a book Building Secure Software: How to Avoid Security Problems the Right Way to be your habit, you can get far more advantages, like add your capable, increase your knowledge about many or all subjects. It is possible to know everything if you like open and read a book Building Secure Software: How to Avoid Security Problems the Right Way. Kinds of book are a lot of. It means that, science e-book or encyclopedia or others. So , how do you think about this guide?

Thomas Obrien:Reading a guide tends to be new life style in this particular era globalization. With examining you can get a lot of information that could give you benefit in your life. Having book everyone in this world could share their idea. Publications can also inspire a lot of people. A lot of author can inspire their own reader with their story or their experience. Not only situation that share in the guides. But also they

write about the data about something that you need example of this. How to get the good score toefl, or how to teach your children, there are many kinds of book that you can get now. The authors on earth always try to improve their ability in writing, they also doing some study before they write to the book. One of them is this Building Secure Software: How to Avoid Security Problems the Right Way.

Sara Kelly:Building Secure Software: How to Avoid Security Problems the Right Way can be one of your nice books that are good idea. Many of us recommend that straight away because this e-book has good vocabulary that could increase your knowledge in terminology, easy to understand, bit entertaining but nevertheless delivering the information. The copy writer giving his/her effort to get every word into delight arrangement in writing Building Secure Software: How to Avoid Security Problems the Right Way although doesn't forget the main point, giving the reader the hottest in addition to based confirm resource details that maybe you can be among it. This great information can drawn you into brand new stage of crucial thinking.

Audrey Mack:Beside this particular Building Secure Software: How to Avoid Security Problems the Right Way in your phone, it might give you a way to get nearer to the new knowledge or data. The information and the knowledge you can got here is fresh from your oven so don't be worry if you feel like an old people live in narrow commune. It is good thing to have Building Secure Software: How to Avoid Security Problems the Right Way because this book offers for you readable information. Do you occasionally have book but you would not get what it's exactly about. Oh come on, that will not end up to happen if you have this in your hand. The Enjoyable set up here cannot be questionable, like treasuring beautiful island. Use you still want to miss this? Find this book in addition to read it from today!

Read Building Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw for online ebookBuilding Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Building Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw books to read online.Online Building Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw ebook PDF downloadBuilding Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw DocBuilding Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw MobipocketBuilding Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw EPubYCJ9PGRI3S7: Building Secure Software: How to Avoid Security Problems the Right Way By John Viega, Gary McGraw