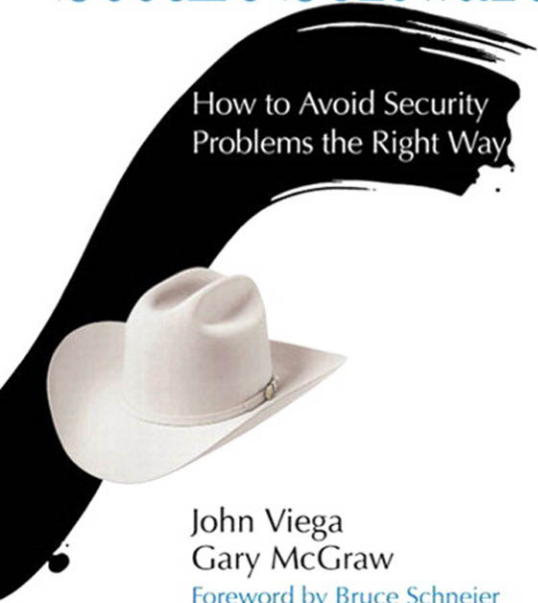


# Building Secure Software



How to Avoid Security  
Problems the Right Way



John Viega  
Gary McGraw  
Foreword by Bruce Schneier

## Advance Praise for *Building Secure Software*

“John and Gary offer a refreshing perspective on computer security. Do it right the first time and you won’t have to fix it later. A radical concept in today’s shovelware world! In an industry where major software vendors confuse beta testing with product release, this book is a voice of sanity. A must-read!”

—*Marcus J. Ranum, Chief Technology Officer,  
NFR Security, Inc. and author of Web Security Sourcebook*

“System developers: Defend thy systems by studying this book, and cyberspace will be a better place.”

—*Fred Schneider, Professor of Computer Science,  
Cornell University and author of Trust in Cyberspace*

“Time and time again security problems that we encounter come from errors in the software. The more complex the system, the harder and more expensive it is to find the problem. Following the principles laid out in *Building Secure Software* will become more and more important as we aim to conduct secure and reliable transactions and continue to move from the world of physical identification to the world of digital identification. This book is well written and belongs on the shelf of anybody concerned with the development of secure software.”

—*Terry Stanley, Vice President, Chip Card Security,  
MasterCard International*

“Others try to close the door after the intruder has gotten in, but Viega and McGraw begin where all discussions on computer security should start: how to build security into the system up front. In straightforward language, they tell us how to address basic security priorities.”

—*Charlie Babcock, Interactive Week*

“Application security problems are one of the most significant categories of security vulnerabilities hampering e-commerce today. This book tackles complex application security problems—such as buffer overflows, race conditions, and implementing cryptography—in a manner that is straightforward and easy to understand. This is a must-have book for any application developer or security professional.”

—*Paul Raines, Global Head of Information Risk Management,  
Barclays Capital and Columnist, Software Magazine*

# **Building Secure Software: How to Avoid Security Problems the Right Way, Portable Documents**

## **Table of Contents**

Contents

Foreword

Preface

Organization

Code Examples

Contacting Us

Acknowledgments

1 Introduction to Software Security

Its All about the Software

Dealing with Widespread Security Failures

Bugtraq

CERT Advisories

RISKS Digest

Technical Trends Affecting Software Security

The ilities

What Is Security?

Isnt That Just Reliability?

Penetrate and Patch Is Bad

On Art and Engineering

Security Goals

# **Table of Contents**

Prevention

Traceability and Auditing

Monitoring

Privacy and Confidentiality

Multilevel Security

Anonymity

Authentication

Integrity

Know Your Enemy: Common Software Security Pitfalls

Software Project Goals

Conclusion

## **2 Managing Software Security Risk**

An Overview of Software Risk Management for Security

The Role of Security Personnel

Software Security Personnel in the Life Cycle

Deriving Requirements

Risk Assessment

Design for Security

Implementation

Security Testing

A Dose of Reality

Getting People to Think about Security

Software Risk Management in Practice

When Development Goes Astray

When Security Analysis Goes Astray

The Common Criteria

Conclusion

## **3 Selecting Technologies**

# **Table of Contents**

Choosing a Language

Choosing a Distributed Object Platform

CORBA

DCOM

EJB and RMI

Choosing an Operating System

Authentication Technologies

Host-Based Authentication

Physical Tokens

Biometric Authentication

Cryptographic Authentication

Defense in Depth and Authentication

Conclusion

## **4 On Open Source and Closed Source**

Security by Obscurity

Reverse Engineering

Code Obfuscation

Security for Shrink-Wrapped Software

Security by Obscurity Is No Panacea

The Flip Side: Open-Source Software

Is the Many-Eyeballs Phenomenon Real?

Why Vulnerability Detection Is Hard

Other Worries

On Publishing Cryptographic Algorithms

Two More Open-Source Fallacies

The Microsoft Fallacy

The Java Fallacy

An Example: GNU Mailman Security

# **Table of Contents**

More Evidence: Trojan Horses

To Open Source or Not to Open Source

Another Security Lesson from Buffer Overflows

Beating the Drum

Conclusion

## **5 Guiding Principles for Software Security**

Principle 1: Secure the Weakest Link

Principle 2: Practice Defense in Depth

Principle 3: Fail Securely

Principle 4: Follow the Principle of Least Privilege

Principle 5: Compartmentalize

Principle 6: Keep It Simple

Principle 7: Promote Privacy

Principle 8: Remember That Hiding Secrets Is Hard

Principle 9: Be Reluctant to Trust

Principle 10: Use Your Community Resources

Conclusion

## **6 Auditing Software**

Architectural Security Analysis

Attack Trees

Reporting Analysis Findings

Implementation Security Analysis

Auditing Source Code

Source-level Security Auditing Tools

Using RATS in an Analysis

The Effectiveness of Security Scanning of Software

Conclusion

# **Table of Contents**

## **7 Buffer Overflows**

What Is a Buffer Overflow?

Why Are Buffer Overflows a Security Problem?

Defending against Buffer Overflow

Major Gotchas

Internal Buffer Overflows

More Input Overflows

Other Risks

Tools That Can Help

Smashing Heaps and Stacks

Heap Overflows

Stack Overflows

Decoding the Stack

To Infinity . . . and Beyond!

Attack Code

A UNIX Exploit

What About Windows?

Conclusion

## **8 Access Control**

The UNIX Access Control Model

How UNIX Permissions Work

Modifying File Attributes

Modifying Ownership

The umask

The Programmatic Interface

Setuid Programming

Access Control in Windows NT

# **Table of Contents**

- Compartmentalization
- Fine-Grained Privileges
- Conclusion

## **9 Race Conditions**

- What Is a Race Condition?
- Time-of-Check, Time-of-Use
  - Broken passwd
  - Avoiding TOCTOU Problems
- Secure File Access
- Temporary Files
- File Locking
- Other Race Conditions
- Conclusion

## **10 Randomness and Determinism**

- Pseudo-random Number Generators
  - Examples of PRNGs
  - The Blum-Blum-Shub PRNG
  - The Tiny PRNG
  - Attacks Against PRNGs
  - How to Cheat in On-line Gambling
  - Statistical Tests on PRNGs
- Entropy Gathering and Estimation
  - Hardware Solutions
  - Software Solutions
  - Poor Entropy Collection: How to Read Secret Netscape Messages
- Handling Entropy
- Practical Sources of Randomness



# **Table of Contents**

Tiny

Random Numbers for Windows

Random Numbers for Linux

Random Numbers in Java

Conclusion

## **11 Applying Cryptography**

General Recommendations

Developers Are Not Cryptographers

Data Integrity

Export Laws

Common Cryptographic Libraries

Cryptlib

OpenSSL

Crypto++

BSAFE

Cryptix

Programming with Cryptography

Encryption

Hashing

Public Key Encryption

Threading

Cookie Encryption

More Uses for Cryptographic Hashes

SSL and TLS (Transport Layer Security)

Stunnel

One-Time Pads

Conclusion

## **12 Trust Management and Input Validation**

# **Table of Contents**

A Few Words on Trust

Examples of Misplaced Trust

- Trust Is Transitive

- Protection from Hostile Callers

- Invoking Other Programs Safely

- Problems from the Web

- Client-side Security

- Perl Problems

- Format String Attacks

Automatically Detecting Input Problems

Conclusion

## **13 Password Authentication**

Password Storage

Adding Users to a Password Database

Password Authentication

Password Selection

- More Advice

- Throwing Dice

- Passphrases

- Application-Selected Passwords

One-Time Passwords

Conclusion

## **14 Database Security**

The Basics

Access Control

Using Views for Access Control

Field Protection

Security against Statistical Attacks

# **Table of Contents**

Conclusion

## **15 Client-side Security**

Copy Protection Schemes

License Files

Thwarting the Casual Pirate

Other License Features

Other Copy Protection Schemes

Authenticating Untrusted Clients

Tamperproofing

Antidebugger Measures

Checksums

Responding to Misuse

Decoys

Code Obfuscation

Basic Obfuscation Techniques

Encrypting Program Parts

Conclusion

## **16 Through the Firewall**

Basic Strategies

Client Proxies

Server Proxies

SOCKS

Peer to Peer

Conclusions

## **Appendix A: Cryptography Basics**

The Ultimate Goals of Cryptography

Attacks on Cryptography

# **Table of Contents**

Types of Cryptography

Symmetric Cryptography

Types of Symmetric Algorithms

Security of Symmetric Algorithms

Public Key Cryptography

Cryptographic Hashing Algorithms

Other Attacks on Cryptographic Hashes

Whats a Good Hash Algorithm to Use?

Digital Signatures

Conclusions

References

Index