SecureSoftware™

eliminating vulnerabilities at the source

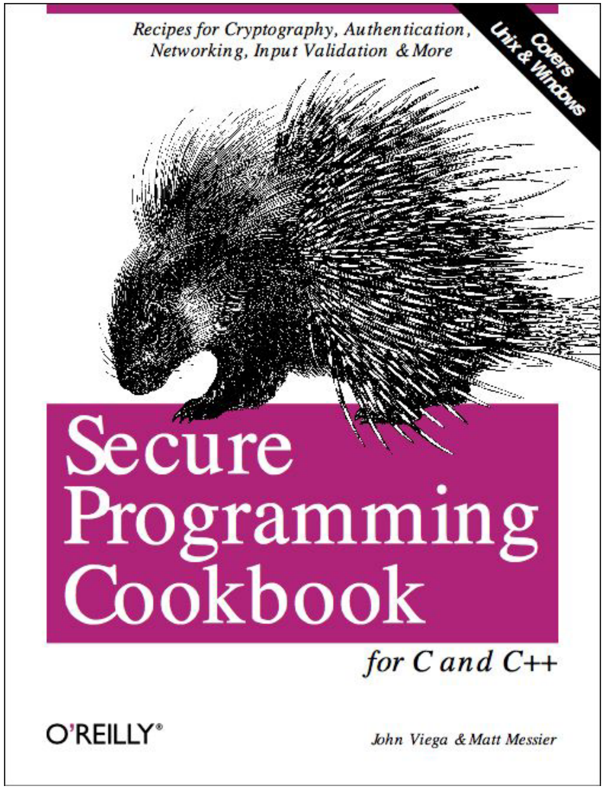
Understanding Software Security

John Viega

Founder & Chief Scientist

Introduction

- Who is Secure Software?
  - Services and technologies for finding security vulnerabilities in software
  - Providing the core technology for the NMCI Product Evaluation Center
- John Viega, Chief Scientist
  - Author of 3 software security books
  - Developed core technologies
- Peter Thimmesch, CEO
  - Peer-to-peer pioneer developer
- Concepts to be discussed
  - Core security issues are in the software, not the network
  - What should you do about the problem?
  - What are the implications for NMCI?


Recipes for Cryptography, Authentication, Networking, Input Validation & More
Covers Unix & Windows
Secure Programming Cookbook
for C and C++
O'REILLY®
John Viega & Matt Messier

Problems are in the software

- An analogy: Airplanes are the software, the security checkpoint is network security
- *Virtually every security problem is due to bugs in software*
- Network security is only a first layer of defense
  - Firewalls only limit communication avenues
  - Intrusion detection only detects against "known" attacks.
- Another analogy: Both the recipe and the chef need to be competent to make a good cookie
- *Producing secure software requires skillful design and coding both*

Problems for everybody

- Consequences vary
  - Denial of service
  - Data theft
  - Data destruction
  - Complete system penetration (ability to install and run new programs)
- Occasionally attacks are highly automated ("Worms")
  - SQL Slammer: Microsoft SQL Server
  - Code Red, Nimda: Microsoft IIS Server
- Dozens of new problems are publicly reported daily
- Bad guys use these and unreported problems when breaking into machines

Economics

- Despite $8.5B spent annually on perimeter security the number of breaches is significantly on the rise
- Cost of recovering from security breaches cost $1.5T worldwide in 2000[1]
- Security breaches cost U.S. businesses with more than 1,000 employees $266B or 2.7% GDP[1]
- Software vendors may eventually face liability, and can currently lose “brand”

“Coding errors in commercial software account for 80% systems penetration. This is clearly a national security issue.”[2]

1) InformationWeek & PricewaterhouseCoopers
2) Air Force CIO John Gilligan, *Information Week* - 3/18/02

SecureSoftware™
eliminating vulnerabilities at the source

Why So Many Coding Errors?

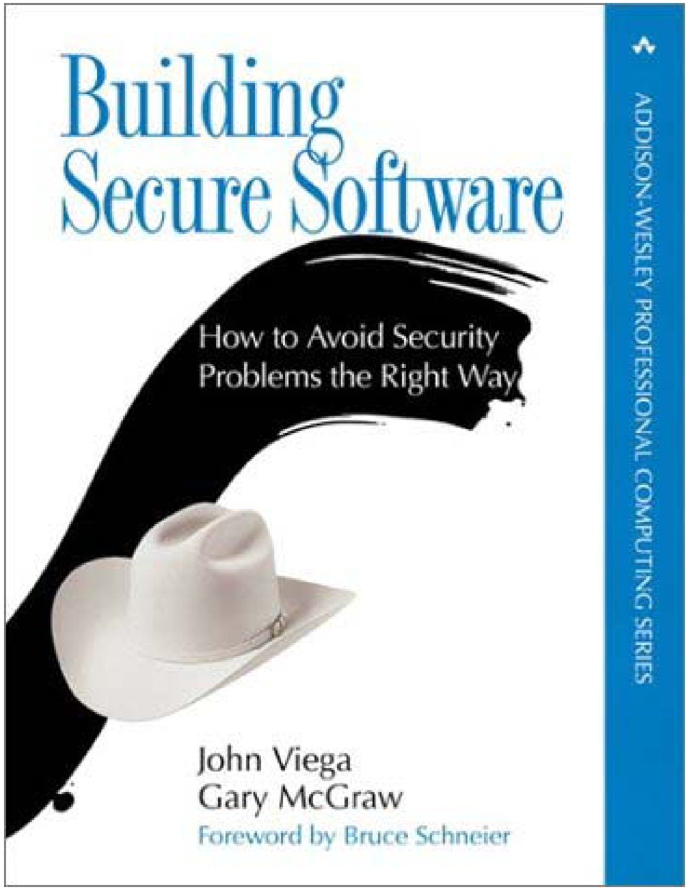
- Software developers:
  - are driven to meet deadlines due to time to market pressure
  - lack security expertise
  - do not have any tools for secure software development
  - wrongly assume perimeter security adequately guards applications
- It's only partially their fault!
  - Software security is a vast, complex topic, and rarely is a high priority in feature-driven development
  - Development is hard enough without needing to be a security expert!

Common Misconceptions

- Fiction: Intrusion Detection solves the problem
  - IDS systems rarely stop suspect connections
  - High false positive rates
  - New vulnerabilities can almost always get through IDS systems
  - Gartner has said IDS will be obsolete by 2005
- Fiction: Java doesn’t have security problems
  - Most problems are in architecture, not implementation
  - Flaws per 1000 lines of code:
    - C: about 3
    - Java: about 1
  - Numbers vary significantly depending on the inherent risks and quality of development
- Fiction: SSL solves our problems
  - I give a talk, “Why SSL isn’t securing your software”
  - The gist: It’s used wrong at least 90% of the time
  - Even so, there’s still plenty of room for other problems

Fixing software development

- Integrate security throughout the development lifecycle (build with security in mind)
- Cultivate software security expertise (e.g., by training)
- Use independent third-party reviews
- Leverage tools for producing secure software (such tools will improve over time)


Building
Secure Software
How to Avoid Security Problems the Right Way
John Viega
Gary McGraw
Foreword by Bruce Schneier
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES