Secure Programming
CS 236
On-Line MS Program
Networks and Systems Security
Peter Reiher

Outline

- Introduction
- Principles for secure software
- Choosing technologies
- Major problem areas
- Evaluating program security

Introduction

- How do you write secure software?
- Basically, define security goals
- And use techniques that are likely to achieve them
- Ideally, part of the whole process of software development
  - Not just some tricks programmers use

Designing for Security

- Often developers design for functionality
  - "We'll add security later"
- Security retrofits have a terrible reputation
  - Insecure designs offer too many attack opportunities
- Designing security from the beginning works better

For Example,

- Windows 95 and its descendants
- Not designed with security in mind
- Security professionals assume any networked Windows 95 machine can be hacked
  - Despite later security retrofits

Defining Security Goals

- Think about which security properties are relevant to your software
  - Does it need limited access?
  - Privacy issues?
  - Is availability important?
- And the way it interacts with your environment
  - Even if it doesn't care about security, what about the system it runs on?

Security and Other Goals

- Security is never the only goal of a piece of software
- Usually not the primary goal
- Generally, secure software that doesn't meet its other goals is a failure
- Consider the degree of security required as an issue of *risk*

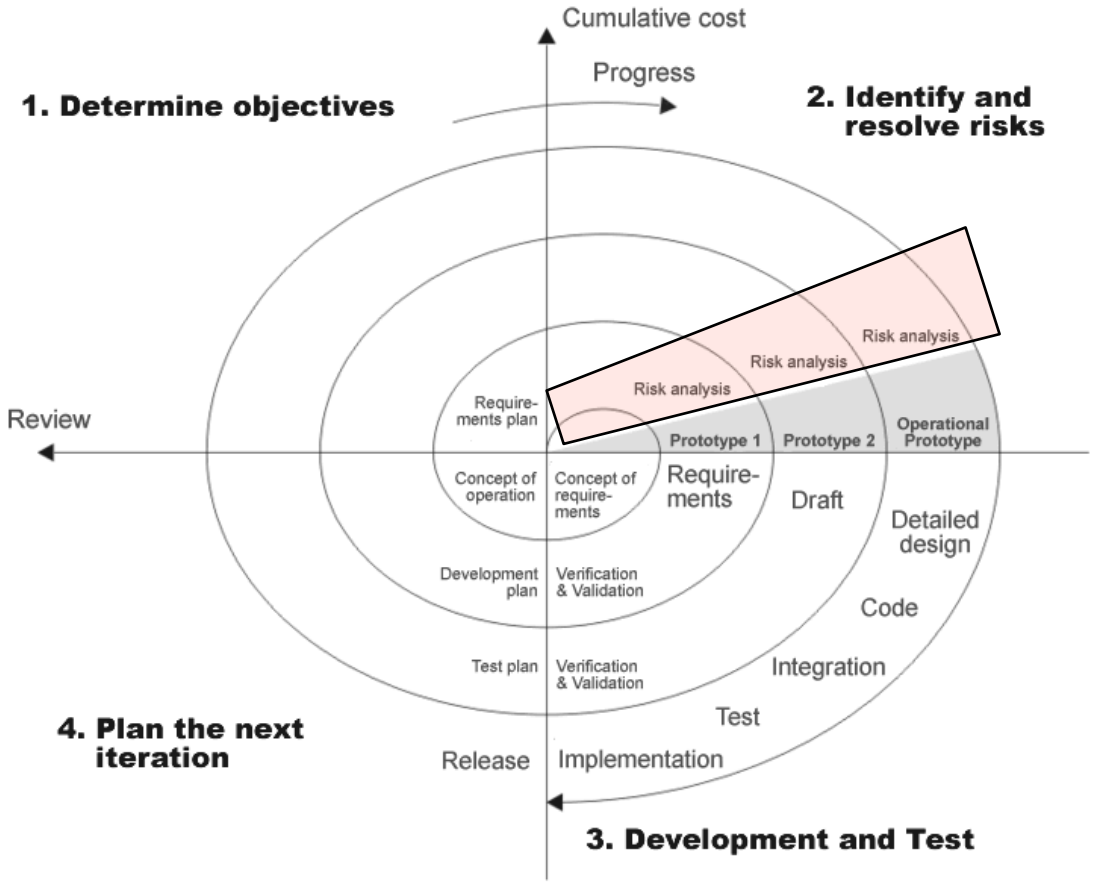
Managing Software Security Risk

- How much risk can this software tolerate?
- What compromises can you make to minimize that risk?
  - Often other goals conflict with security
  - E.g., should my program be more usable or require strong authentication?
- Considering tradeoffs in terms of risks can clarify what you need to do

Risk Management and Software Development

- Should consider security risk as part of your software development model
- E.g., in spiral model, add security risk analysis phase to the area of spiral where you evaluate alternatives
- Considering security and risks early can avoid pitfalls later
- Returning to risk when refining is necessary

Incorporating Security Into Spiral Model of SW Development



Include security in the risks you consider

At all passes through the spiral

But How Do I Determine Risk?

- When you're just thinking about a big new program, how can you know about its risks?
- Well, do the best you can
  - Apply your knowledge and experience
  - Really think about the issues and problems
  - Use a few principles and tools we'll discuss
- That puts you ahead of 95% of all developers
- You can't possibly get it all right, but any attention to risk is better than none

Design and Security Experts

- Someone on a software development team should understand security
  - The more they understand it, the better
  - Ideally, someone on team should have explicit security responsibility
- Experts should be involved in all phases
  - Starting from design