



Health Net<sup>®</sup>

Application security in the SDLC

Practical Strategies



# Contents

- Application security today
- Compliance pressures and industry solutions
- Application security for the “perfect world”
- Practical strategies for the “real world”



# Application security today

In 2008,

- most data breaches were caused by outsiders.\*
- most breaches were discovered by third parties.\*
- nearly *all* compromised records were from online.\*

Application security problems span industries

- No one is immune.

Secure application development is a challenge

- Continuous lifecycle approach is required.
- *No one-time silver bullet approach will work.*



# Application security threats

- Data loss, disclosure or corruption
- Denial of service
- Accomplice to an attack



# Compliance pressures and industry efforts

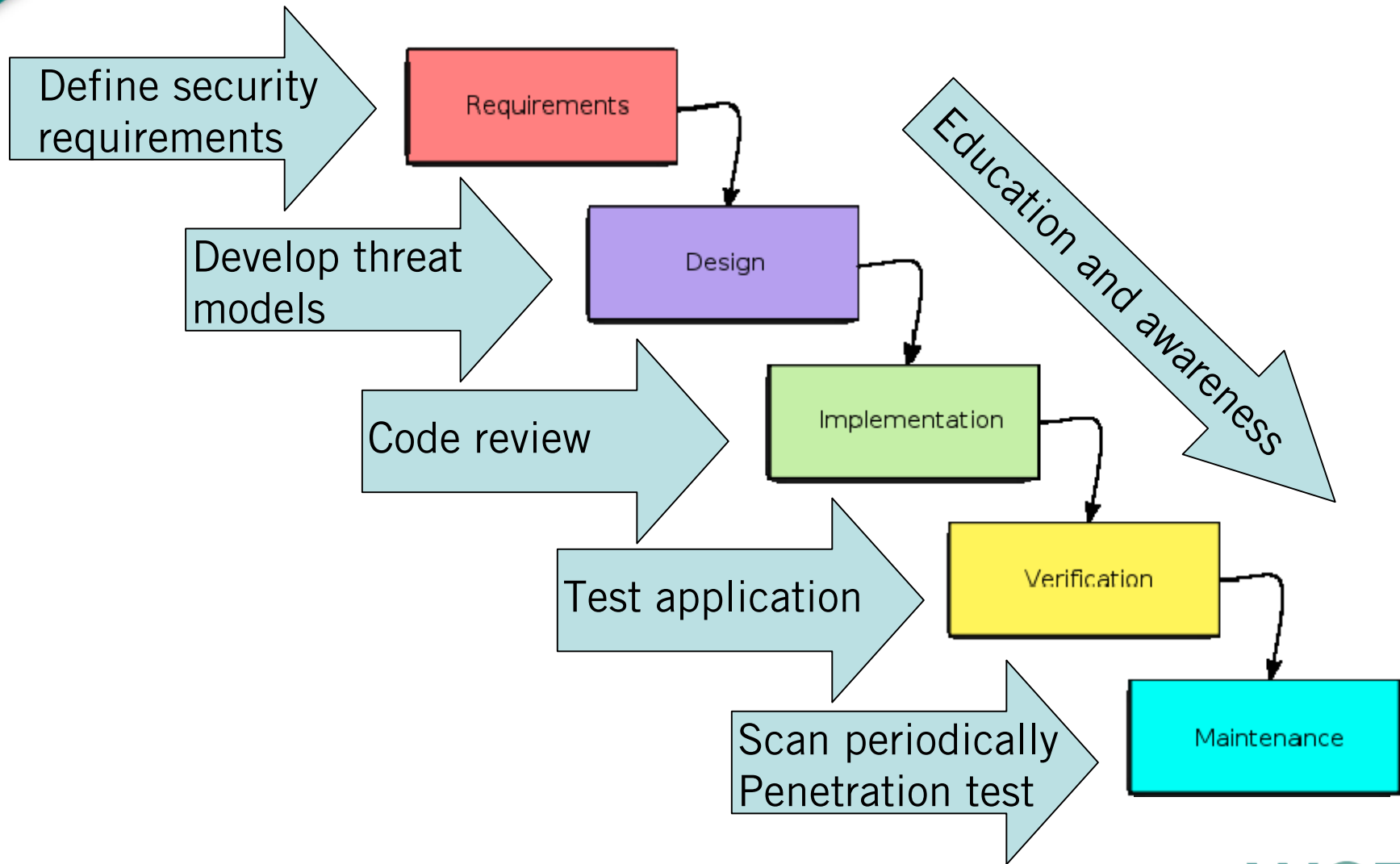
Compliance comes in many flavors and acronyms

- HIPAA, SOX, FISMA, PCI, FERPA (and more)

Improvement efforts

- Open Web Application Security Project (OWASP)
- Web Application Security Consortium
- Microsoft
- SANS

# Application security in the “perfect world”



Few of us live in a “perfect world”

- You may not be involved until code is near to or currently in production
- Resources are limited
- Time is of the essence



So, where do you go from here?

*Before embarking on security improvements, ensure you've got your code under control.*

- Release management is vital
- Your best efforts to correct problems will be washed away if you cannot control your code.

*As you begin, foster application security awareness at all levels of the development organization*

- Plant the seeds of awareness
- Most people tend to want to do the right thing



## Practical strategies (cont)

*You've got to figure out what to fix*

- Test applications with scanner
- Assess access control design
- Determine compliance with company policy
- Benchmark against industry best-practices

### *Talk about it*

- Communicate the results within the organization
- Be honest but keep risk in perspective
- Prioritize issues and seek ideas for solutions

### *Develop a time-phased plan to get well*

- Near term (within 6 months)
  - Consider / budget for additional resources
  - Remediate code
  - Consider training options / budget for robust formal training
  - Develop requirements to comply with security standard

### *Develop a time-phased plan to get well*

- Mid term (between 6 - 12 months)
  - Continue communication within your organization
  - Develop and perform test cases to evaluate application
  - Perform routine scanning of production code
  - Scan applications prior to new releases
  - Implement / enforce secure code standards
  - Implement / enforce code review
  - Perform / budget for independent audit

### *Develop a time-phased plan to get well*

- Long term (1 year and beyond)
  - Implement “best-practices” projects
  - Continue testing production applications (monthly, if possible)
  - Continue testing applications prior to release
  - Continue code reviews (consider automated)
  - Improve / update coding standards
  - Conduct independent audits yearly

*Communicate and celebrate successes!*

- There is no silver bullet or “one-size-fits-all” plan
- Cost and schedule will always be a challenge
- Use practical strategies to “eat the elephant one bite at a time.”