Secure Code Development

ISACA South Florida

7th Annual WOW! Event

Agenda

i. Background
ii. Building a Business Case for Secure Coding
iii. Top-Down Approach to Develop Secure Code
iv. Developing a Secure Code Initiative
v. Integrating Best Coding Practices into the Software Development Life Cycle (SDLC)
vi. Secure Code Awareness and Training for Developers
vii. Conduct an Independent Secure Code Audit
viii. Auditing Software After a Breach
ix. Summary
x. References

Secure Code Development

Background

- What is software security?
  - Protecting software against malicious attack so that the software continues to function correctly
  - The ability of the software to ensure the Confidentiality, Integrity, and Availability of the sensitive data on the system

Secure Code Development

Background

- Measuring software security
  - Dependability
  - Trustworthiness
  - Resilience
  - Conformance

Software Assurance

Secure Code Development

Background

- Perspective Case Study: **Target Data Breach**

  "Target said 40 million credit and debit card accounts had been affected by a data breach that happened between November 27th and December 15th, 2013."

  "The Consumer Bankers Association estimated that the cost of card replacement for its members has reached $172 M."

  "When the final tally is in, Target's breach may eclipse the theft at TJX, which compromised more than 90 million records."

- The Associated Press

Secure Code Development

Building a Business Case for Secure Coding

- The cost of insecure software
  - ✓ Software consumers are demanding software assurance
  - ✓ Fixing bugs and flaws is very costly after implementation
  - ✓ Compliance with technology security standards

Secure Code Development

Top-Down Approach to Develop Secure Code

- Make enterprise security the responsibility of leaders at a governance level
- Develop a secure code initiative and create a secure coding team (not just developers)
- Empower IT leadership with a budget and the political capital to succeed
- Use a benchmark to evaluate the performance of the organization's secure code initiative. For example, Building Security In Maturity Model (BSIMM)

Secure Code Development

Developing a Secure Code Initiative

- Select a framework to help the organization formulate and implement a strategy for software security. For example, Open Software Assurance Maturity Model (OpenSAMM)
- Adopt a secure coding standard
- Integrate secure coding best practices into the SDLC
- Implement an awareness and training program for secure code development
- Integrate collaborative testing tools and techniques to facilitate code reviews (automated processes)

Secure Code Development

Integrating Best Coding Practices into the SDLC

- **Software Development Models:**
  - ❖ Waterfall model
  - ❖ Spiral model
  - ❖ Rapid application development
  - ❖ Agile development
  - ❖ And others

Secure Code Development

Integrating Best Coding Practices into the SDLC

- **Agile Development Model:**
  - **Agile Manifesto (2001):** Promotes software development following these principles:
    - ✓ Individuals and interactions over processes and tools
    - ✓ Working software over comprehensive documentation
    - ✓ Customer collaboration over contract negotiation
    - ✓ Responding to change over following a plan

Secure Code Development

Integrating Best Coding Practices into the SDLC

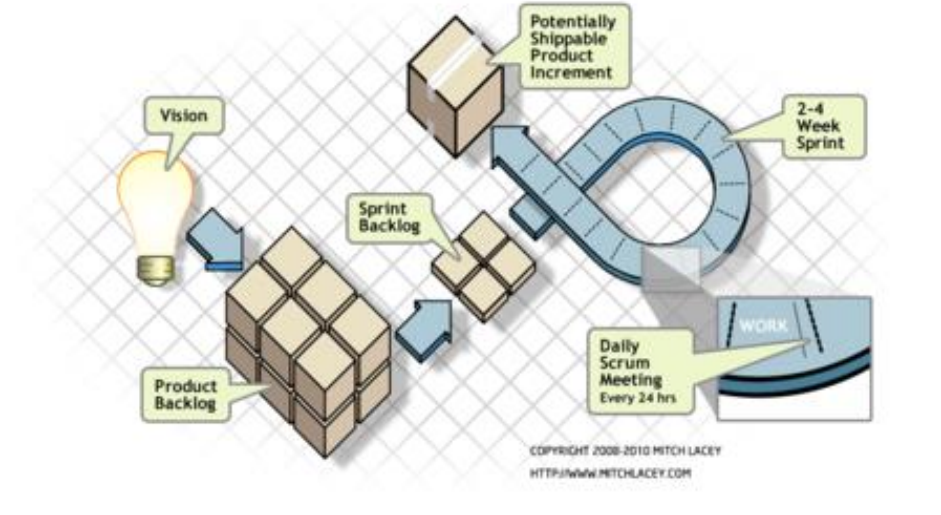
- **Agile Development Model**
  - Agile Methods Security Concerns:
    - ✓ Agile is about building software quickly
    - ✓ Move fast and iterate
    - ✓ Respond to feedback
    - ✓ Emphasis on velocity and business value
    - ✓ Deliver software before it is finished

Secure Code Development

Integrating Best Coding Practices into the SDLC

➢ Agile Scrum

- Scrum is an agile method for project management
  - ✓ It is a TEAM based collaborative approach
  - ✓ Iterative and incremental
  - ✓ Always focus on delivery



Secure Code Development

Integrating Best Coding Practices into the SDLC

➢ Agile Scrum

- Sprints are the basic units of development in Scrum
  - ✓ Usually from 2 to 4 weeks long
  - ✓ The release date of a product determines the number of Sprints
  - ✓ Sprint planning sessions assign tasks to developers
  - ✓ At the end of each sprint there is a customer (product owner) demo
  - ✓ Requirements may change after the demo phase

Secure Code Development

Integrating Best Coding Practices into the SDLC

- **Agile Scrum**
  - Scrum Master Role
    - Facilitate sprint planning, retrospective and sprint demos
    - Assist the product owner with keeping the backlog groomed
    - Ensure cross-team coordination
    - Reach out to the larger company network for impediment removal
    - Etc.

Secure Code Development

Integrating Best Coding Practices into the SDLC

- **What about software security?**

"Incorporate Security activities into all phases of the SDLC process, from initiation to disposal, regardless of the software development method used."

Agile Method: Development life cycle is divided into "increments" or "iterations", in which each of these increments touches on each of the conventional phases of development.

Secure Code Development

Integrating Best Coding Practices into the SDLC

- **What about software security?**

"Incorporate Security activities into all phases of the SDLC process, from initiation to disposal, regardless of the software development method used."

Traditional Methods: Development life cycle is divided into systems analysis and requirements definition, systems design, development, integration and testing, acceptance, installation, deployment, evaluation, and disposal.

Secure Code Development

Integrating Best Coding Practices into the SDLC

- **Software Requirements Definition**
  - Consider security requirements, not just functional requirements
    - ✓ Authentication
    - ✓ Role Based access controls
    - ✓ Separation of duties
    - ✓ Automated data integrity checks

Secure Code Development

Integrating Best Coding Practices into the SDLC

- **System Design**
  - ➢ Perform software architecture risk analysis
    - ✓ Review exception handling for application overrides
    - ✓ Evaluate:
      - ❑ Functional and security requirements
      - ❑ Component interfaces
      - ❑ Component communications and dependencies
      - ❑ Component responsibilities

Secure Code Development

Integrating Best Coding Practices into the SDLC

- **System Implementation**
  - Perform code reviews before application goes into deployment
    - ✓ Pay attention to “backdoors” in the source code
    - ✓ Look out for shortcuts to get the product out the door
    - ✓ Don’t accept the comment “But the user would never do that!

Secure Code Development

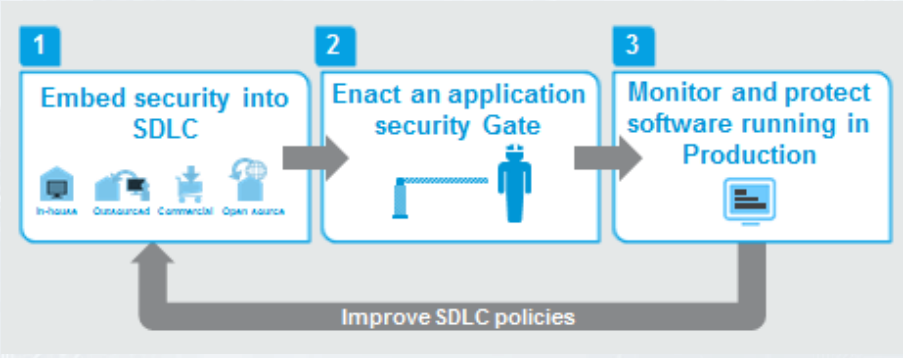
Integrating Best Coding Practices into the SDLC

- **System Deployment**
  - Review separation of duties during deployment
  - Examine backup procedures and documentation
  - Test passwords used on the sandbox environment
  - Monitor application behavior for unexpected results

Secure Code Development

Integrating Best Coding Practices into the SDLC

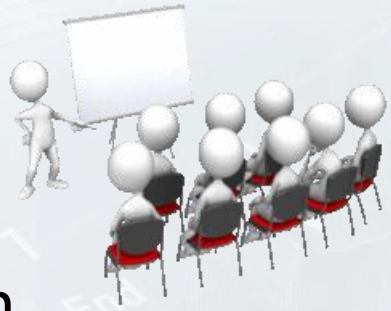
- **System Maintenance**
  - Conduct regular code reviews
  - Evaluate change management controls
  - Review major changes to components of the application



Secure Code Development

Secure Code Awareness and Training for Developers

- Awareness + Training = Education
- Embed training into the SDLC
- Education should be an ongoing process
- Measure the effectiveness of this program
- Outcome of Training = Accountability



Secure Code Development

Conduct an Independent Secure Code Audit

- **Purpose:**
  - **Security Flaws and Bugs:** Discover flaws in the programming that may be lead to security exploitations. For example:
    - ✓ **Unsecure and unvalidated input or output** - Cross-site scripting, SQL injection
    - ✓ **Improper Exception Handling** - Debugging and error messages
    - ✓ **Insecure communication protocols**- Clear-text and unauthenticated protocols
    - ✓ **Broken or incomplete authentication controls** - User ID manipulation
    - ✓ **Flawed session management** - Session or Cookie hijacking
    - ✓ **Bounds checking** - Buffer or Integer overflows
    - ✓ **Weak storage encryption** - Broken or insecure encryption algorithms

Secure Code Development

- **Auditing Software after a Breach**

  - **Forensic investigative and analytical skills and abilities are needed**
    - ✓**Technical skills**
      - Building a digital audit trail
      - Understand computer fraud techniques
      - Understand information collected from various computer logs
      - Understand the inner workings of web servers, firewalls, attack methodology, security procedures & penetration testing

Secure Code Development

- **Auditing Software after a Breach**

  - **Forensic investigative and analytical skills and abilities are needed**
    - ✓**Conduct a review of:**
      - Computer Incident Response Plan and its performance after a successful cyber attack
      - Chain-of-custody process
      - Information Security Policies and Procedures
      - Secure Code Awareness and Training for Developers
      - Organizational and legal protocols for incident handling
      - Other elements of the Secure Code Development Program

Secure Code Development

Summary

- ✓ Software assurance implies dependability, trustworthiness, resilience, and conformance
- ✓ Consumers are demanding better software security
- ✓ Security is the responsibility of leaders at a governance level
- ✓ Integrate secure coding best practices early
- ✓ Embed software architecture and code risk analysis into the SDLC
- ✓ Develop a secure code awareness and training program for all stakeholders
- ✓ Conduct independent secure code audits
- ✓ Assume the organization will have a security breach , so be prepared

Secure Code Development

References

ISO/IEC 9126

Special Publication 800-64

Q & A

THANK YOU!