IMENTUM

Secure System Development

Håkan Engvall
Imentum Systems AB
www.imentum.se
http://www.linkedin.com/in/engvall

1

IMENTUM

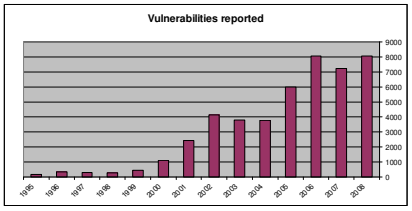
Educational Objectives

- After this class you should have a good understanding of:
  - how security requirements can be managed in system development projects
  - what parts of the development process are effected and how
  - what methods to use evaluated security requirements in a finished system
- The class will also give an overview of the following:
  - what are the technical threats to IT systems
  - what methodologies exists to write "secure" code
  - what tools are available to write good quality (and secure) code

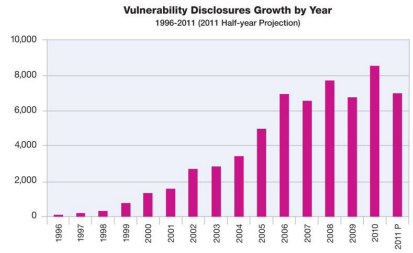
2

IMENTUM

Increasing numbers of vulnerabilities and incidents



Source: www.cert.org



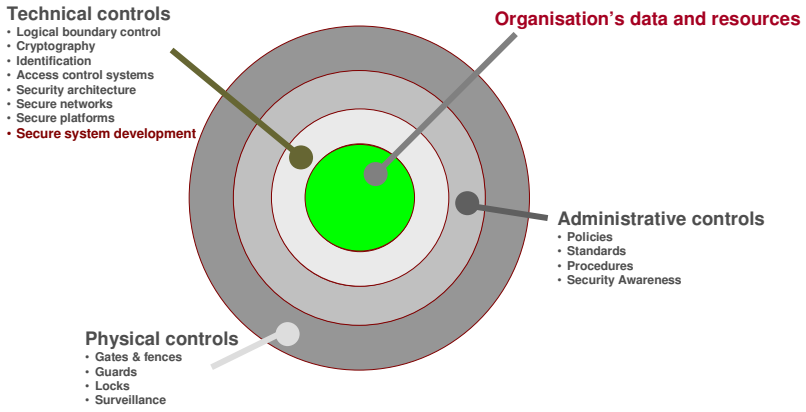
Source: IBM X-Force team

- ”Surveys show that commercial software has about 5-15 error per 1000 lines of code.”
- ”...1-5 security flaws per 1000 lines of code...”
- Previous studies of quality in software development.
  - Design 21% ROI
  - Implementation 15% ROI
  - Test 12% ROI

3

IMENTUM

Part of the bigger picture



4

IMENTUM

Three choices for your development strategy

Reuse existing solutions and/or components

- Is it possible to add on to existing solution?
- How secure is the base for new requirements?

Buy (parts or a complete system)

- Do pre-made components exist? A complete system?
- How secure are these parts/components and systems?

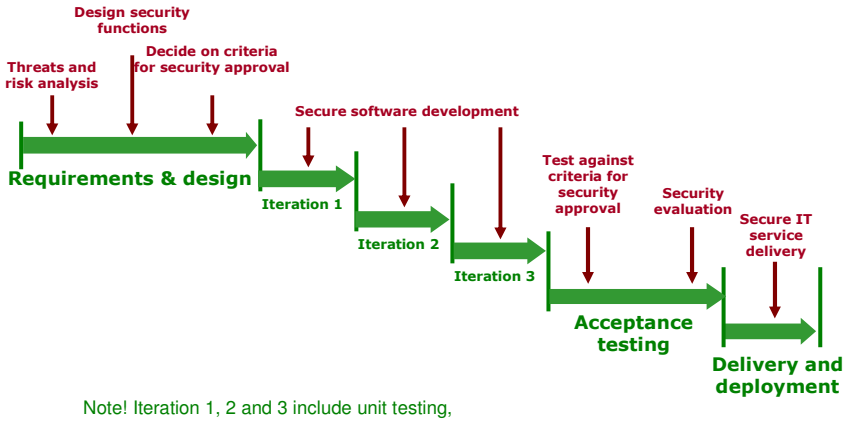
In-house development

- Start a system development project
- Needs to use secure technologies and methods

5

IMENTUM

Classical development lifecycle



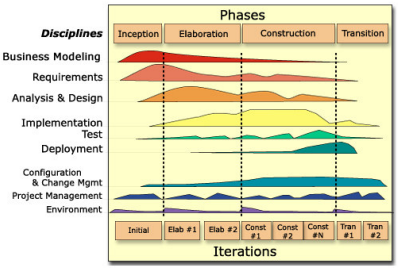
Note! Iteration 1, 2 and 3 include unit testing, module testing, integration testing, etc.

6

IMENTUM

RUP phases

- Inception
  - Define the project
  - Estimates of costs and risks
- Elaboration
  - Requirement specification (functional and non-functional)
  - Identify project risks and mitigating controls
- Construction
  - Design, implementation, tests
- Transition
  - Acceptance testing
  - Delivery and deployment



7

IMENTUM

Information security management systems

- 27000 — Overview and vocabulary
- 27001 — Requirements
- 27002 — Code of practice
- 27003 — Implementation guidance
- 27004 — Measurement
- 27005 — Information security risk management
- 27006 — Requirements for audit and certification

8

IMENTUM

Information security management systems

0. Introduction
1. Scope
2. Terms and definitions
3. Structure of this standard
4. Risk Assessment and Treatment
5. Security policy
6. Organizing information security
7. Asset management
8. Human resources security
9. Physical and environmental security
10. Communications and operations management
11. Access control
12. Information systems acquisition, development and maintenance
13. Information security incident handling
14. Business continuity management
15. Compliance

ISO/IEC 27001 and 27002

9

IMENTUM

What do ISO 27000 say about system development

- Chapter 12 defines the follow objectives:
  - To ensure that security is an integral part of information systems.
  - To prevent errors, loss, unauthorized modification or misuse of information in applications.
  - To protect the confidentiality, authenticity or integrity of information by cryptographic means.
  - To ensure the security of system files.
  - To maintain the security of application system software and information.
  - To reduce risks resulting from exploitation of published technical vulnerabilities.

Source: ISO/IEC 27002:2005 Information security management systems, chapter 12

10

IMENTUM

Mapping between RUP & 27000

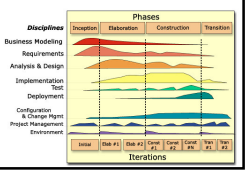
	Inception	Elaboration	Construction	Transition
To ensure that security is an integral part of information systems				
To prevent errors, loss, unauthorized modification or misuse of information in applications				
To protect the confidentiality, authenticity or integrity of information by cryptographic means				
To ensure the security of system files				
To maintain the security of application system software and information				
To reduce risks resulting from exploitation of published technical vulnerabilities				

11

IMENTUM

Project management

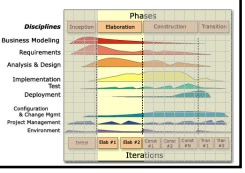
- Security aspects should not be allowed to be forgotten during any phase of the project
  - check lists should be implemented at project milestones
- The flow from requirement and specification, through design and implementation, to acceptance testing and delivery has to hold even for security aspects
- Change management with traceability from change request or error report to change and vice versa
- Think security throughout the system's whole life cycle
  - plan for the time after the system development project



12

Design & architecture

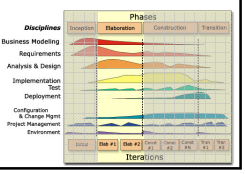
- Security requirements are seldom tangible
  - common wording ”no unauthorised person should be able...”
- Proper construction of security mechanisms are not simple
  - ”home grown” solutions are not recommended
  - if possible use established frameworks
- It’s often difficult to design test for security functions
  - you are going to test things that shouldn’t work
- Plan early for delivery and deployment
  - IT service operation needs system descriptions, how to read logs, act on alarms, etc.



13

Requirement management

- The requirement specification must include security requirements
- Gather the business security requirement through
  - Analysis
    - Business requirement
    - Legal and regulatory requirements
    - Threat based requirements
  - Prioritise
    - Risk analysis
    - Cost/benefit calculations
- Security requirements shall cover
  - Confidentiality
  - Integrity
  - Availability
  - Traceability
- Write explicit security requirements

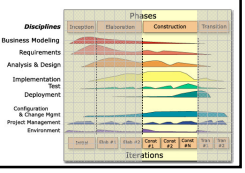


14

Architecture and design

IMENTUM

- Architecture with security in depth
  - boundary control (logical and physical)
  - infrastructure (networks)
  - operating systems
  - applications
- Reuse (if possible) mechanisms for:
  - authentication
  - cryptography
  - traceability (logging)
  - etc.
- Create (if necessary) framework for these mechanisms
  - Service oriented architecture (SOA)??

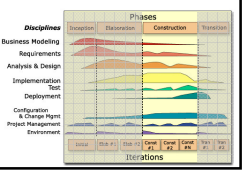


15

Implementation

IMENTUM

- Create guidelines for developers
- Examples
  - Least privilege principle
  - Checks against trojans and other malware
  - Use code analysis and coverage tools
  - Common best practices guidelines
  - Defensive programming
    - always have an explicit branch for all possible input
    - check input and output conditions (and invariants)
- Change management decision must fit the organisation's and project's security requirements

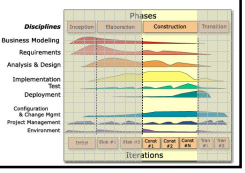


16

Change management

IMENTUM

- Every change shall be traceable to cause, version, time and date, and person
- List of all included components
- List of all tools need to build the system (incl. version information)
- Complete build and install instructions

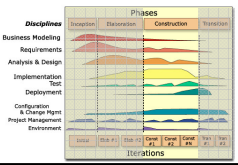


17

Tools to write (more) ~~secure~~ better code

IMENTUM

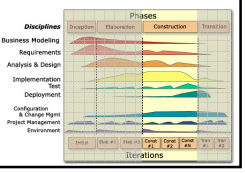
- Methods
  - code review
  - pair programming (XP-method)
  - test driven development
  - ...
- Tools
  - static code analysis
    - http://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis
  - dynamic code analysis
    - http://en.wikipedia.org/wiki/Dynamic_program_analysis
  - code coverage
    - http://en.wikipedia.org/wiki/Code_coverage



18

Testing

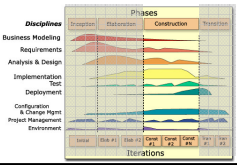
- Create a test strategy including security testing
- As with all testing, security testing is a continues process
- It's important to also include business processes, people and technology in the testing. Too often we just test the technical parts.
- We need to think about the whole life cycle of the system or product
  - Make sure that security requirements are explicit
  - Specify, design, implement, test, deliver and deploy the system through a secure development methodology
  - Perform security testing regularly, even after delivery
- Note!
  It is important to test the system together with business processes, system administration processes and continuity procedures



19

Testing, cont.

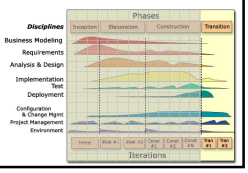
- Involve the test team from the start of the project
- "Think Evil. Be Evil. Test Evil."
  - Automate attacks with scripts and other tools
  - Input data that is outside the specified ranges
  - Deny access to files, register keys, database tables or columns
  - Test with non-administrator accounts
- Know your enemy and know yourself
  - What tools and techniques are attackers using?
  - What tools are your testers using?



20

Deployment

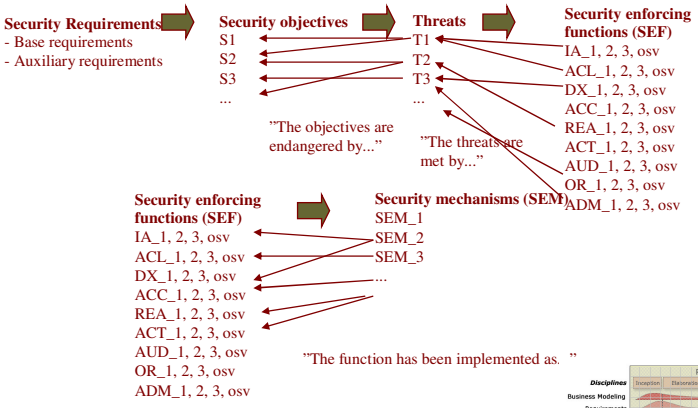
- Requirement specification on infrastructure
  - backups, time synchronisation, log management etc.
  - needs to be done early in the project
- Deliverables to IT service operation
  - system documentation
  - documentation from tests, checklists
  - compare with accreditation or certification
  - procedures for patch management
- Deliverables to software maintenance
  - Remaining known problems
  - procedures for patch management

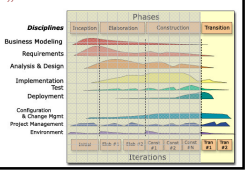


21

Accreditation/certification

Means that one verifies that all requirements have been implemented in the delivered system.





22

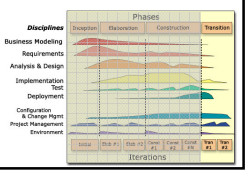
Common Criteria

- ISO Standard 15408
- Used for evaluating and certifying IT products
- A product is described with
  - Security Functional Requirements (SFR)
  - Security Assurance Requirements (SAR)
- SFR defines what security features exists in the product
- SAR defines how the product is developed, deployed and used
- Products are evaluated according to different assurance levels (EAL1 – EAL7)
- Most product certifications are at EAL3 or EAL4

23

Operation or development?

- Business continuity planning (BCP)
  - Ultimately the system owner is responsible
  - Should system development or IT service operation plan and design?
- Penetration testing
  - System development knows the whole design and implementation
  - IT service operation knows the operational environment



24

IMENTUM

Common faults and attacks

25

IMENTUM

Architecture and design

- Too powerful user accounts are used
- End-to-end security isn't used
  - open up for man-in-the-middle
- Sensitive data is sent in clear text
- Encryption functions are used in bad ways
  - most common mistake is to not delete keys properly
- Identity and access control are performed on the client side
- Not using randomisation to prevent replay attacks

26

Implementation

IMENTUM

- Not checking input data (injection)
- Cross Site Scripting (XSS)
- Defects in authentication and session handling
- Defects in access control functions
- Cross site request forgery
- Errors in configuration
- Unprotected cryptographic storage
- Failure to restrict URL access
- Not using transport layer protection
- Unvalidated redirects and forwards

OWASP Top Ten Application Security Risks

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

27

IT operation

IMENTUM

- Leaving defaults accounts with default passwords active
- IT operation environment hasn't been cleaned from development tools and example code
- Making it possible to download scripts and other source code
- Download hashed passwords
- Not keeping up with patch management

28

IMENTUM

Security Metrics for System Development

29

IMENTUM

Acquisition & Implementation

- Number and percentage of system with completed security analysis
- Security mechanisms coverage for
  - confidentiality
  - integrity
  - availability
  - traceability
- Consultation between
  - business unit
  - developers
  - security experts

30

Installing & Accrediting Solutions

IMENTUM

- Number & percentage of deployed systems with
  - certification (tested and deemed compliant)
  - accreditation (sign-off and risk accepted)
- Accreditation / Certification includes
  - system documentation
  - Service Level Agreements (SLA)
  - operational instructions (daily, weekly, monthly, etc. tasks)
  - continuity planning
  - ...

31

Educational Objectives

IMENTUM

- After this class you should have a good understanding of:
  - how security requirements can be managed in system development projects
  - what parts of the development process are effected and how
  - what methods to use evaluated security requirements in a finished system
- The class will also give an overview of the following:
  - what are the technical threats to IT systems
  - what methodologies exists to write "secure" code
  - what tools are available to write good quality (and secure) code

32