Integrating Software Assurance and Secure Programming Concepts and Mindsets into an Undergraduate Computer Science Program

Striving to Achieve the Goals of the SEI/CERT Software Assurance Curriculum Project (Undergraduate)

Steve Hadfield
U.S. Air Force Academy, Department of Computer Science

Realization

In an outcome-based curriculum,
some outcomes need to be purposefully developed across courses and years.

Result

A retrospective, outcome-based look at an existing curriculum (Felder & Brent)

Key Cross Curricular Initiative

Software Engineering Discipline	• Needs Analysis, Requirements Elaboration, Design • Testing Rigor, Quality Assurance
Ethical, Legal, Social Issues	• Moral Frameworks & Decision Making • Ethical Codes (IEEE, ACM, Software Engineering)
Research Skills	•Literature Review, Framing/Scoping Topics, Hypotheses • Investigation, Support of Conclusion, Reporting
Communications Skills	• Oral Presentations • Written Communications
Team Work	• Team Building, Team Maintenance • Pair Programming, Four-Five Member Team Dynamics
Security & Software Assurance	• Secure Programming • Cyber Security

Security & Software Assurance

SEI/CERT SwA Curriculum

- Computer Science I
- Computer Science II
- Intro to Computer Security
- Software Security Engineering
- Software Quality Assurance
- Software Assurance Analytics
- Software Assurance Capstone

USAFA Computer Science

- Computer Science I
- Computer Science II
- Computer Security & Information Warfare
- Software Engineering I
- Software Engineering II

Security & Software Assurance Initiative Sophomore Year

Computer Science I - Intro to Programming

- Input interpretation validation, array bounds checking
- Integer overflow, error/exception handling, file I/O issues

Computer Science II – Data Abstraction

- Pre- and post-conditions, more advanced debugging
- Testing & debugging techniques, reinforce CS I topics

Computer Organization & Architecture

- Data type overflow, divide-by-zero, round-off error
- Stack overflows

Security & Software Assurance Initiative
Junior Year

Programming Paradigms
- Memory allocation/deallocation, termination conditions
- Stack/buffer overflows and protections, type safety

Operating Systems
- Deadlock issues, race conditions, system calls
- Signals, file system security

Databases & Web Programming
- Defense against SQL injection attacks
- Cross site scripting attacks

Networks
- Secure protocols, wireless encryption, Man-in-the-Middle attacks
- Adversarial view of protocols, network access control

Security & Software Assurance Initiative Senior Year

Languages & Machines (compilers & language theory)

- Type checking mechanisms, array bounds checking mechanisms
- Translation to machine language

Computer Security & Information Warfare

- Security & threat models
- Range of security strategies and techniques

Software Engineering I

- Security requirements, security analysis of system design, risk management
- Formal test plans/procedures/reports
- Integration/system/regression/smoke/stress/security testing

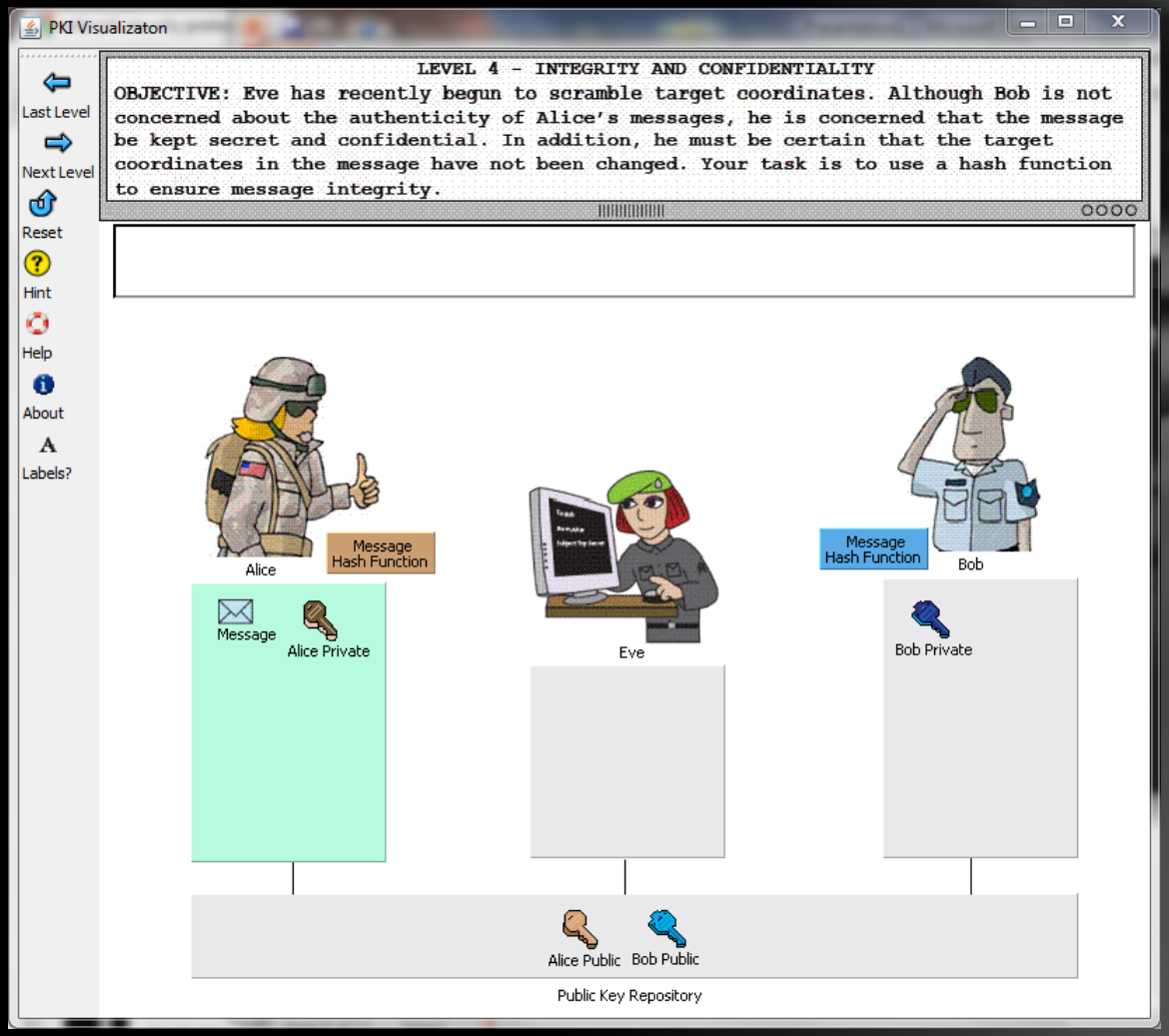
Software Engineering II

- Introduction to Formal Methods
- Reengineering & forward engineering

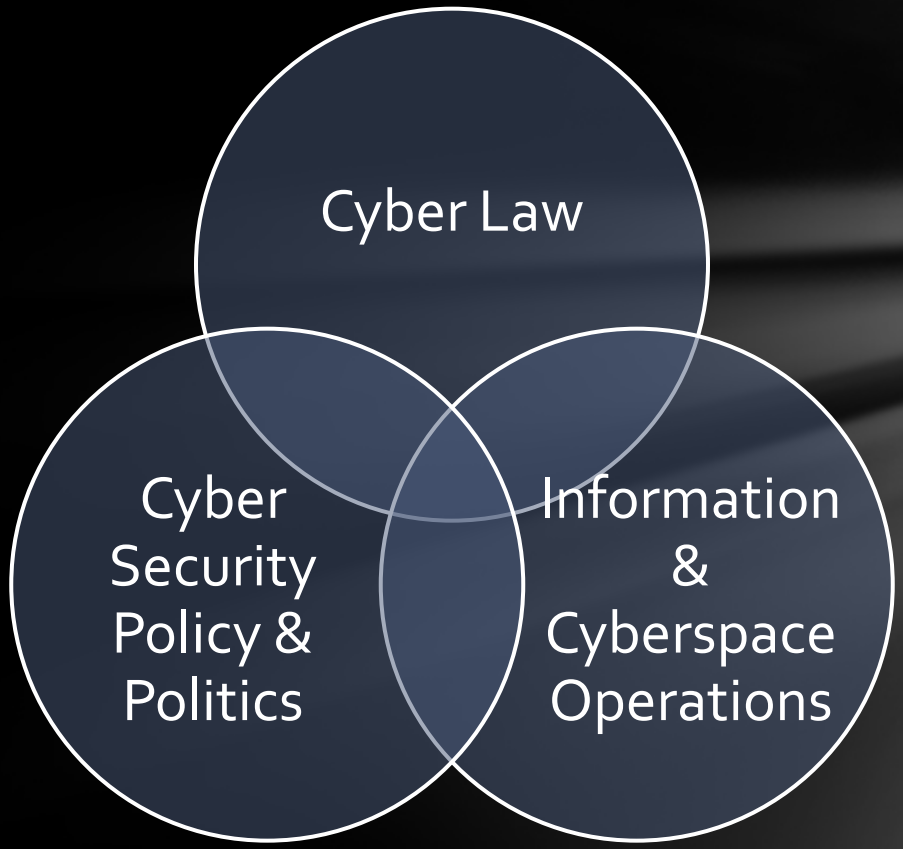
Software Assurance & Security for ALL

- Algorithmic Reasoning
  - Input Validation
  - Exception Prevention
  - Requirements Analysis
  - Incremental Build/Test
- Cyber Security
  - Information Security
  - Cryptography
  - Cyber Warfare & Crime
  - Offensive Cyber Ops
  - Defensive Cyber Ops

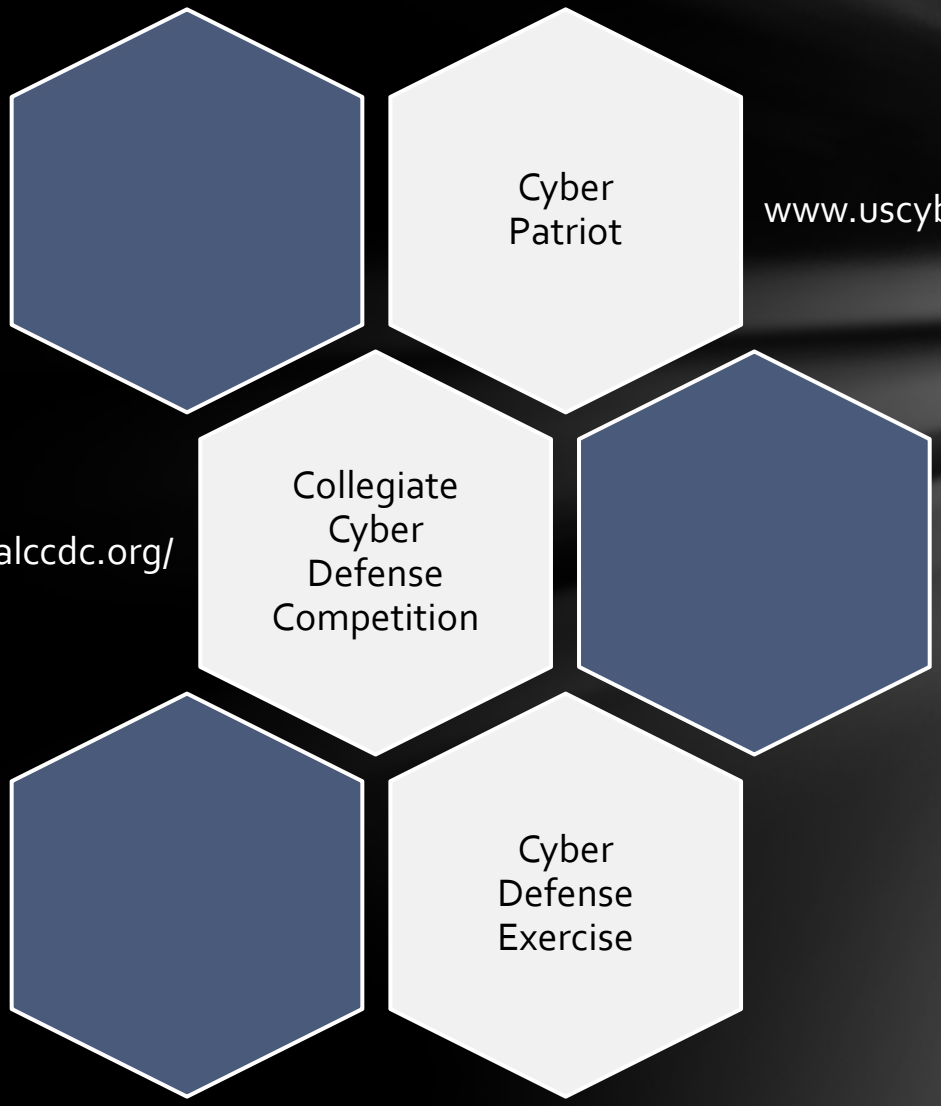
Public Key Infrastructure



Enrichment Activities
Interdisciplinary Courses



Enrichment Activities
Defensive Competitions



www.uscyberpatriot.org

www.nationalccdc.org/

Vectors

Professionals

- Comp Sci, Info Sys, Info Tech, MIS
- Curricular & pedagogical resources

General Awareness

- Personal awareness & defense
- Bigger issues – enterprise, national, global

Specialization

- Defense is the 'hard job'
- Funding for developing experts

Questions?

Steve Hadfield

Steven.Hadfield@usafa.edu