Cybersecurity Curricular Guidelines 2017

Matt Bishop
Dept. of Computer Science
University of California at Davis
1 Shields Ave.
Davis, CA 95616-8562

phone: +1 (530) 752-8060
email: mabishop@ucdavis.edu
web: http://seclab.cs.ucdavis.edu/~bishop

Background

- What should a “cybersecurity professional” know and what skills should they have?
- Disagreement due to nature of cybersecurity
  - Used in many different roles
  - Encompasses many disciplines
- Indeed, many different definitions of cybersecurity

Joint Task Force Sponsors

- Association for Computing Machinery (ACM)
- IEEE Computer Society (IEEE-CS)
- Association for Information Systems Special Interest Group on Security and Privacy (AIS SIGSEC)
- International Federation for Information Processing Technical Committee on Information Security Education (IFIP WG 11.8)

Funders

- US National Science Foundation (Award 1623104)
- ACM Education Board
- Intel Corporation
- US National Security Agency's CNAP Curriculum Development Effort (RFI-2017-00022) and Grant H98230-17-1-0219

Joint Task Force Members

- **Matt Bishop**, University of California Davis, co-chair
- **Diana Burley**, The George Washington University, co-chair
- Scott Buck, Intel Corp.
- Joseph J. Ekstrom, Brigham Young University
- Lynn Futcher, Nelson Mandela Metropolitan University
- David Gibson, United States Air Force Academy
- Elizabeth K. Hawthorne, Union County College
- Siddharth Kaza, Towson University
- Yair Levy, Nova Southeastern University
- Herbert Mattord, Kennesaw State University
- Allen Parrish, United States Naval Academy

Joint Task Force

- Assembled by ACM, IEEE Computer Society, AIG Special Interest Group on Security, IFIP WG 11.8
- Goal: to develop cybersecurity curricular guidelines for undergraduate programs that emphasize different areas of specialization
  - Not curriculum development!
  - Institutions can use it as a basis for developing curricula, certification requirements, though

Available now!
http://cybered.acm.org

CYBERSECURITY CURRICULA 2017

Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity

A Report in the Computing Curricula Series
Joint Task Force on Cybersecurity Education

Association for Computing Machinery
IEEE computer society
ASSOCIATION FOR INFORMATION SYSTEMS
ifip

- First set of global cybersecurity guidelines
- Developed with the assistance of more than 325 contributors across 35 countries
- Our vision: leading resource of comprehensive curricular content for global institutions developing cybersecurity offerings across a wide range of disciplines

What Is Cybersecurity?

"A computing-based discipline involving technology, people, information, and processes to enable assured operations. It involves the creation, operation, analysis, and testing of secure computer systems. It is an interdisciplinary course of study, including aspects of law, policy, human factors, ethics, and risk management in the context of adversaries."

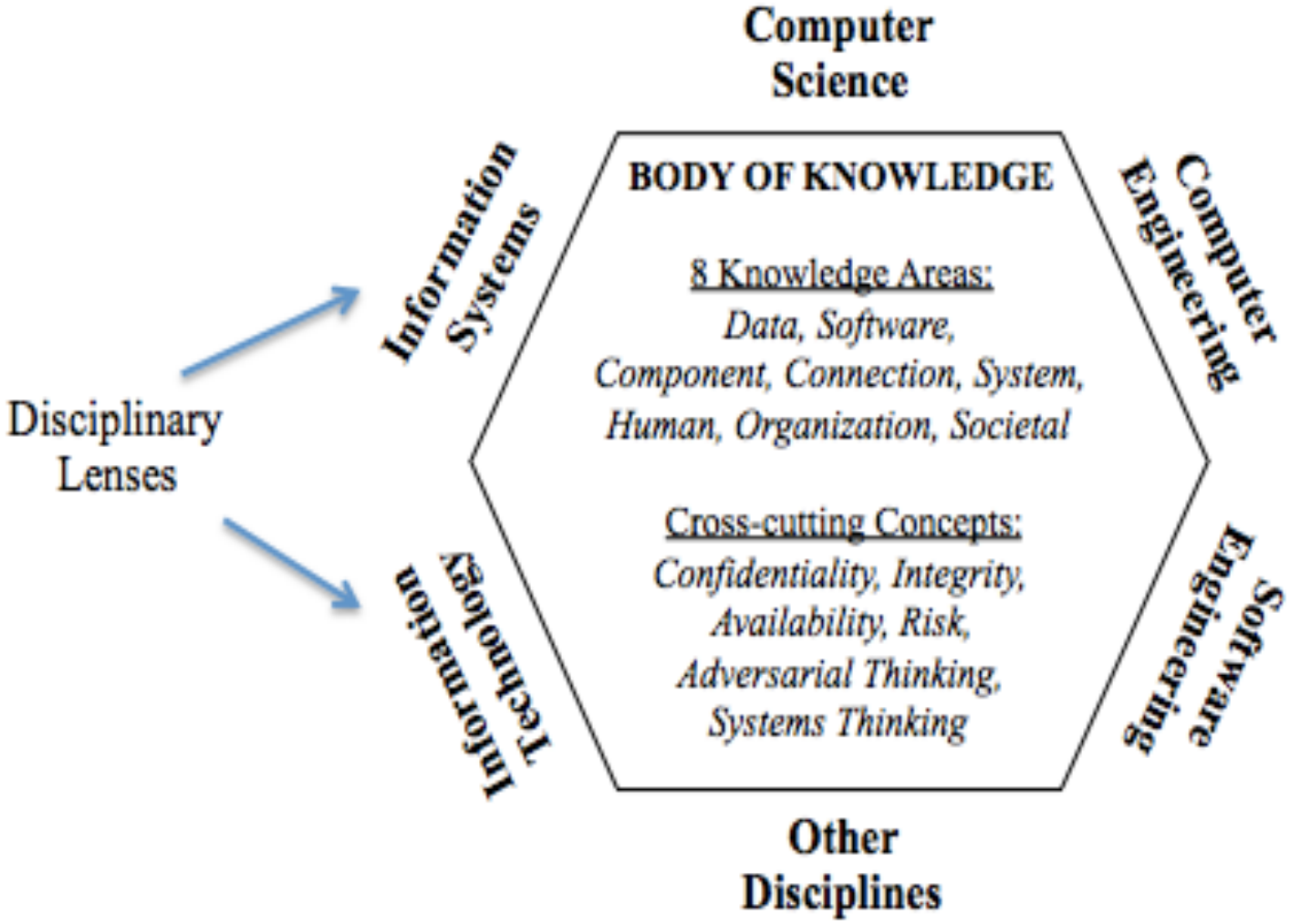
— CSEC 2017 report, p. 10

Bases for Guidelines

Cybersecurity education programs should:

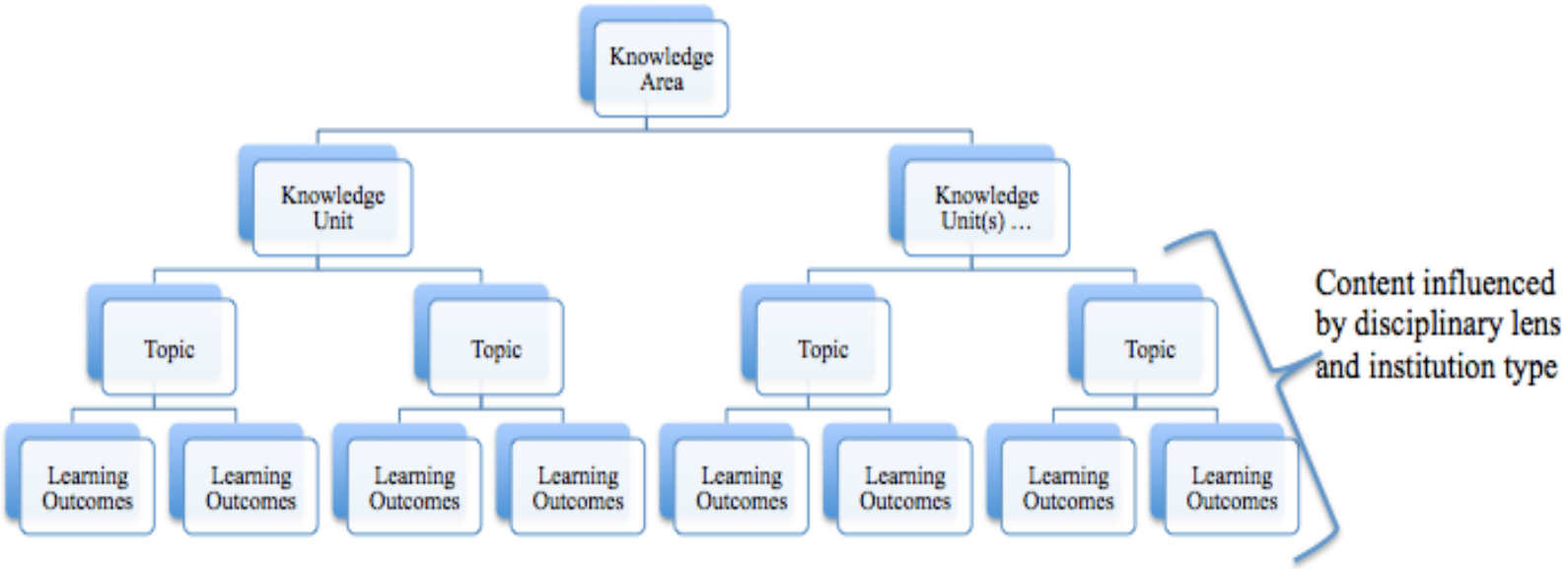
- Be based on core knowledge and skills;
- Have a computing-based foundation
- Teach concepts applicable to a broad range of cybersecurity expertise;
- Emphasize ethical obligations and responsibilities; and
- Be flexible so programs can tailor their curriculum to any specialized needs

The Model


Computer Science
BODY OF KNOWLEDGE
8 Knowledge Areas:
Data, Software, Component, Connection, System, Human, Organization, Societal
Cross-cutting Concepts:
Confidentiality, Integrity, Availability, Risk, Adversarial Thinking, Systems Thinking
Information Systems
Computer Engineering
Information Technology
Software Engineering
Disciplinary Lenses
Other Disciplines

Knowledge Areas

- The subject matter of cybersecurity
- Each KA composed of Knowledge Units
  - These describe the set of topics and what students should know about them

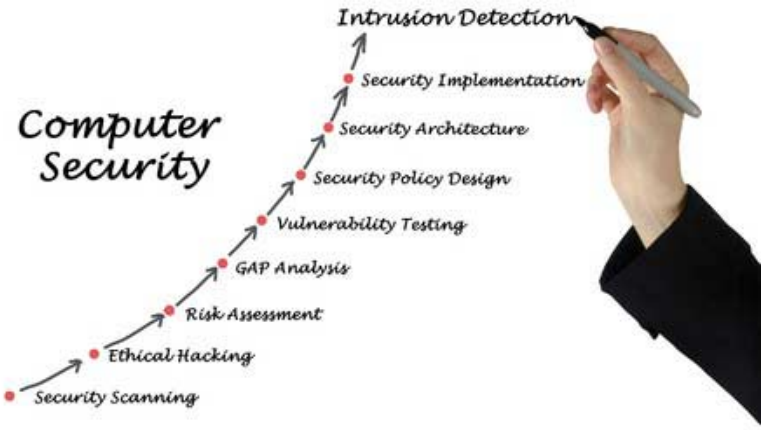


Knowledge Areas

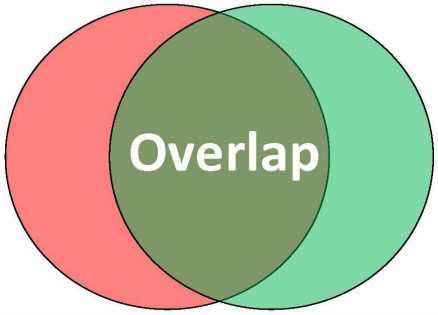
- KAs meet 3 criteria:
  1. The area is important for multiple disciplines;
  2. The area provides a tool for understanding or exploring cybersecurity ideas; and
  3. The material in the area can be learned in varying levels of detail and understanding over time.

The Knowledge Areas

- Data Security
- Software Security
- Component Security
- Connection Security
- System Security
- Human Security
- Organizational Security
- Societal Security



Overlap



Note a Knowledge Unit may be put into more than one KA

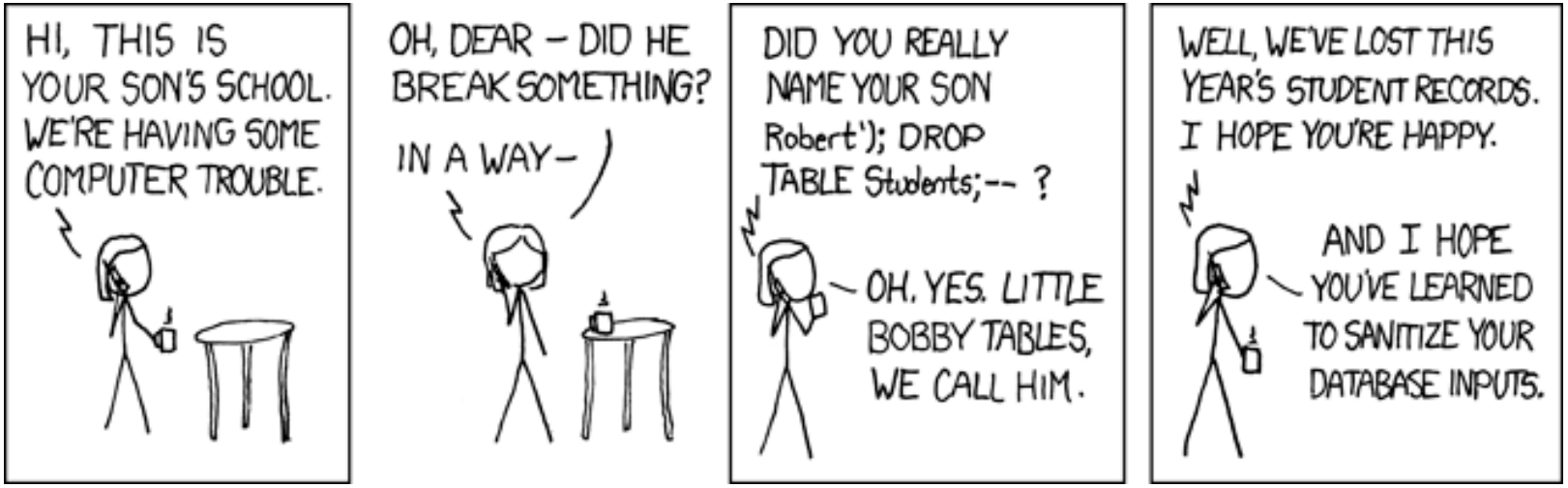
Example: design of a library that grabs and processes packets

- Software security KA: assurance question is how packets are protected and properly read
- System security KA: composition of components, as library interface ties network to system

Example: Software Security

CSEC2017 Knowledge Units:

- Fundamental design principles
- Practice
- Documentation



Example: Software Security

Original (in CSEC2017 Draft)

- KU: fundamental design principles
  - Separation (of domains)
  - Isolation
  - Encapsulation
  - Least Privilege
  - Simplicity (of design)
  - Minimization (of implementation)
  - Fail Safe Defaults Fail Secure
  - Modularity
  - Layering
  - Least Astonishment
  - Open Design
  - Usability
  - End-to-End Security
  - Defense in Depth

Working Group Modifications

- KU: fundamental design principles

  Simplicity principles:
  - Economy of mechanism
  - Minimize shared mechanisms
  - Least astonishment

  Restrictive principles:
  - Least privilege
  - Fail-safe defaults
  - Complete mediation
  - Separation
  - Minimize trust

  Methodology principles:
  - Open design
  - Layering
  - Abstraction
  - Complete linkage
  - Design for iteration

Example: Other Software Security KUs

- KU: Design
  - Derivation of Security Requirements
  - Specification of security requirements
  - Software/Security Development Life Cycle
  - Programming languages/type-safe languages
- KU: Implementation
  - Validating input and checking its representation
  - Using APIs correctly
  - Using security features
  - Checking time, state relationships
  - Handling exceptions, errors properly
  - Programming Robustly
  - Encapsulating structures, modules
  - Taking environment into account
- KU: Analysis and Testing
  - Static, dynamic analysis
  - Unit testing
  - Integration testing
  - Software testing
- KU: Deployment and Maintenance
  - Configuring
  - Patching and the vulnerability life cycle
  - Checking environment
  - DevOps
  - Decommissioning/Retiring
- KU: Documentation
  - Installation documents
  - User guides, manuals
  - Assurance documentation
  - Security documentation
- KU: Ethics
  - Ethical issues in software development
  - Social aspects of software development
  - Legal aspects of software development
  - Vulnerability disclosure
  - What, when, why to test

Example: Software Security Essentials

- Fundamental Design Principles; Least Privilege, Open Design, and Abstraction
  - Discuss the implications of relying on open design or the secrecy of design for security
  - List three principles of security
  - Describe why each principle is important to security
  - Identify the needed design principle

CYBER ESSENTIALS

Example: Software Security Essentials

- Security requirements and the roles they play in design
- Implementation issues
- Static, dynamic analysis
- Configuring, patching
- Ethics, especially in development, testing, and vulnerability disclosure

Cross-Cutting Concepts

- Framework for making connections among KAs, unifying underlying ideas
- Fundamental to students' abilities to understand core ideas through any disciplinary lens
- They span most, if not all, of the KAs

The Concepts

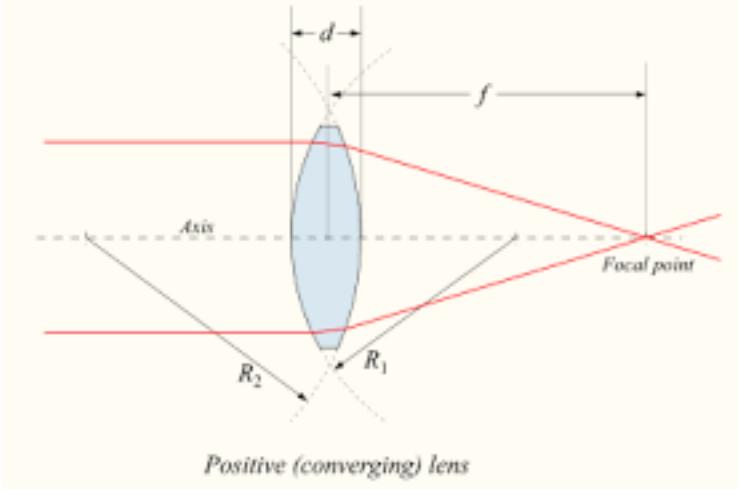
- Confidentiality: property defined by rules controlling spread of information
- Why?
  - Key component of data security, systems security, organizational security
  - Also component of human security in guise of privacy
  - And societal security from the combination of all these

The Concepts

- Integrity
  - Key component: assurance
- Availability
  - Accessibility not enough; must meet any stated QoS
- Risk
  - Function of probability of threat being realized and damage incurred should this happen
- Adversarial Thinking
  - Requires understanding what threats will compromise entity, how to make them happen
- Systems Thinking
  - Considers interplay among technical, social constraints

Disciplinary Lens

- KAs are common to all cybersecurity
- Depth, approach that folks are expected to know varies depending on how they will use that knowledge
- We use ACM disciplines



The Disciplines

- Computer Science
- Computer Engineering
- Information Systems
- Information Technology
- Software Engineering
- Other Disciplinary

Guidelines and Professional Practice

- Linked through seven application areas
- Workforce frameworks can codify their bodies of knowledge by going from the application areas back to the model
  - Then extract both core knowledge, cross-cutting concepts they find appropriate
  - View them through the appropriate disciplinary lens
- These areas allow the definition of competency levels for each area

Application Areas

- Public policy
- Procurement
- Management
- IT security operations
- Software development
- Enterprise architecture
- Research

What Can You Do?

- Exemplars!
  - Mapping courses to CSEC2017 to show how a variety of institutions, programs cover the KA essentials and some subset of the KUs
  - Mapping job requirements to application areas and topics in the KUs to tie them into what employers (you) should expect from people applying for particular jobs
- These will be provided on the cybered.acm.org web page

Conclusion

- CSEC2017 is a *basis* for curricula, and is ***not*** itself a curriculum
- CSEC2017 still evolving

A Contrarian View

The future, according to some scientists, will be exactly like the past, only far more expensive

– John Sladek