CERIAS Tech Report 2002-68
Defining a Curriculum Framework in Information Assurance and Security
by Melissa Dark
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Defining a Curriculum Framework in Information Assurance and Security

James Davis
Information Assurance Center
Department of E CPE
Iowa State University
Ames, Iowa
davis@iastate.edu

Melissa Dark
Center for Research in Information Assurance and Security
Purdue University
West Lafayette, Indiana
dark@cerias.purdue.edu

1. Introduction

In this paper, we describe a community effort to identify the common body of knowledge (CBK) for computer security curricula. Academicians and practitioners have been engaged in targeted workshops for the past two years, producing the results given here. The long-term objective for the project is to develop a curriculum framework for undergraduate and graduate programs in Information Assurance (IA). The framework includes: identification of broad areas of knowledge considered important for practicing professionals in information assurance, identification of key learning objectives for each of these areas, identification of a body of core knowledge and skills that all programs should contain, and a model curriculum including scope and sequence. The framework's development has been facilitated by workshops and working groups of leading information assurance educators. The goal is to produce document similar to the Joint IEEE Computer Society/ACM Task Force document (1) "Model Curricula for Computing" (Computer Science Volume) which will then be widely distributed for comment and dissemination. We anticipate that the framework will be used to guide the development of shared instructional materials, classroom instruction, and the assessment of individuals and programs.

The focus for this paper is the design of the curriculum framework and the identification of the common body of knowledge. One of the interesting challenges is the breadth of the Information Assurance field. There is a tendency to view IA as strictly a subset of computer science, however many of the issues that professionals address require knowledge and skills drawn from traditionally non-computer disciplines. IA is truly a multidisciplinary endeavor, blending topics that span the disciplines of computer science, computer engineering, mathematics, management information systems and business, political science, and law[1]. Additionally, key processes used by IA professionals (e.g., vulnerability assessment) require a deep understanding of how important concepts in each of these disciplines are connected to each other.

The rationale for the project is based in the need to develop a consensus on core IA skills and knowledge. The demand for Information Technology (IT) professionals stemming from turnover plus growth has been pegged in various references at around 600,000 open positions per year (3).

[1] This first step was focused on the more familiar computer science and computer engineering topics. Educators and practitioners from related disciplines are engaged in the project, and content from those disciplines will be included in as the work progresses.

While IT is of course broader than IA, it is generally believed that IA positions comprise a large percentage of the IT shortfall. There is an urgent need to significantly increase the number of graduates who are prepared for careers in the IA fields. A major barrier to meeting this challenge is that few Universities currently offer a comprehensive IA educational program; furthermore, sufficient numbers of experienced faculty to ramp up such an effort does not exist. In a testimony given to the US House of Representative Committee on Science (4) on February 11, 1997, Professor Eugene Spafford from Purdue University presented results from a survey he conducted indicating that there were only 12 faculty members nationwide with significant teaching and researching assignment in Information Assurance. In 2003, we are able to identify only a few score institutions offering more than a single course in network security or cryptography[2].

Given the growing need for graduates educated in computer security and the current lack of a capacity to meet that need, there is a premium placed on leveraging existing expertise by sharing instructional materials for core concepts. This will succeed on the scale needed only if there is an accepted IA curriculum framework in place.

Fortunately, there exists a helpful body of work to build from. One of the key resources is the CNSS training standards for information assurance (2). These documents provide a set of learning objectives for training IA professionals, and can additionally become a good content map for a college courses (several Universities have mapped their graduate courses to the various CNSS standards). Additional resources include the proceedings from WECS (workshop on education in computer security), the "Green Book"[3], SANS short courses, curriculum materials from the COE schools, and many other resources. One of the major challenges is coalescing existing instructional material and bringing stakeholders together in a shared vision of a model curriculum.

2. Curriculum Design Philosophy

Curriculum design and development means many things to many people. This is especially true in education where individuals have tacit understanding of curriculum design, development, and enactment. For the purpose of this project, we turned to the curriclum and instruction literature to establish a working definition that could serve as a guide for discussion and guide understanding of the task at hand and our work. It should be noted that this work has really just begun. Therefore, the curriculum perspectives provided below will continue to guide our work as we move forward.

Curriculum design is concerned with making decisions about the **scope, organization**, and **sequence** of the **content** at the macro level (Smith & Ragan, 1999). **Content** then can be

[2] One metric is the Centers of Academic Excellence in Information Assurance Education program sponsored by the National Security Agency. As of spring 2002, only 36 universities have been identified as having significant undergraduate education or graduate education and research activities.
http://www.nsa.gov/isso/programs/nietp/index.htm

[3] Dr. Corey Shou, National Center of Academic Excellence in Information Assurance Education, Idaho State University, http://security.isu.edu/

considered as the topics to be taught (what should be taught?) **Scope** is a question of how much students should know; to what degree should students be taught this depends upon the degree of understanding/knowledge that you intend them to have upon completion. **Organization** is a question of how to sequence the topics (there are a variety of organization strategies: prior knowledge, job-function, super-ordinate concepts, etc). Finally, **sequence** is the suggested ordering of content based on answers to the three prior questions.

The output of curriculum design varies according to the impetus for and uses of the curriculum design and development effort. The first goal of this project is to produce a document that defines the common body of knowledge in Information Assurance, i.e., what should be taught in Information Assurance program (content). A second goal of this project is to identify key learning outcomes for each of these areas, i.e., what students should know and be able to do (scope) and corresponding performance metrics,i.e., indicators that will serve as evidence of what students know and can do.

With regard to content, this group was seeking to define the core curriculum where core would be viewed as the intersection of various programs. We recognize that different programs will not only have different content, but even different emphases within the core. Furthermore, the group recognized that Information Assurance is multi-disciplinary in nature, including but not limited to disciplines such as psychology, sociology, political science, law, computer science, computer engineering, and management . The multi-disciplinary nature means that what students should know and be able to do will vary across discplines and will require that we establish stronger involvement of experts from related discplines who have not been involved to date. The group also recognized that what students should know and be able to do will vary by the orientation of the specific program and the type(s) of career or advanced schooling being prepared for. Given that, the group felt that we could produce a working document that defined the content, i.e., the common body of knowledge across all discplines and types of programs, but that meaningful definition of scope would need to be more detailed and granular according to program type.

We have utilized a logic model approach and specficially the backward design model (5) to guide the process of defining the core curriculum. The backward design model is derived from the fundamental systems/program logic model whereby antecendents, transactions, and outcomes are logically linked in an apparent and systematic way (Figure 1)

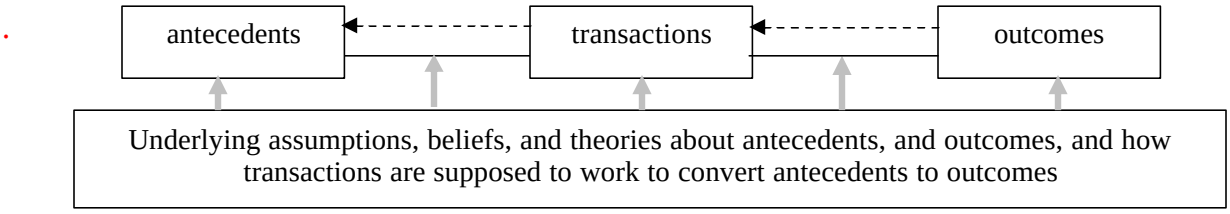

Figure 1: Logic Model

The solid arrows from left to right indicate how a program is supposed to work when operational; the dotted arrows from right to left indicate how the program should be planned. When applied to educational curricula, outcomes are a descriptive representation of what

students should know, understand, and be able to do as evidenced by formal and informal assessment. Transactions then are the learning experiences and instruction that we expect will produce the desired results. The backward design model is a curriculum design (i.e., planning) model that works from right to left and consists of three steps; 1) identifying desired results, 2) determining acceptable evidence, and 3) planning learning experiences and instruction. Steps one and two are used to describe outcomes and step three is used to describe transactions. The goal of our curriculum framework project is to define IA outcomes, those things that students should know and be able to do as information assurance practitioners.

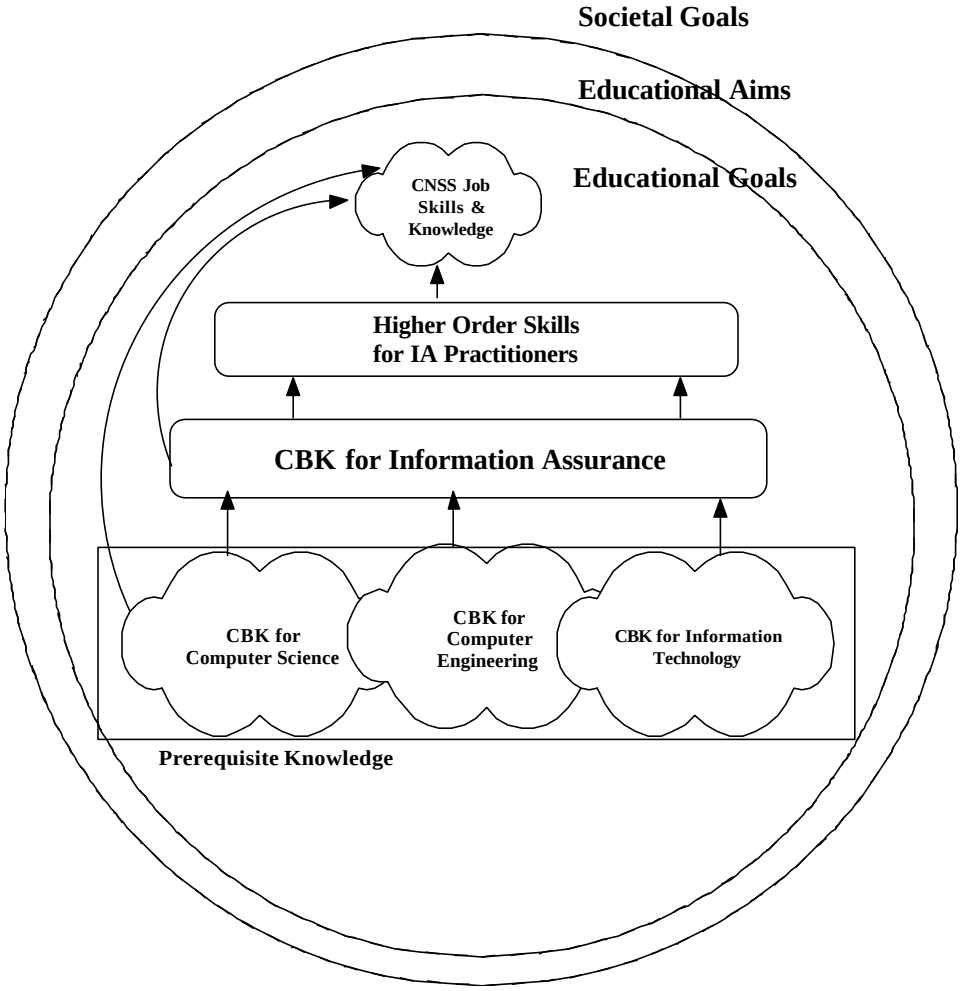

Figure 2. Curriculum Framework

3. Framework Structure

The overarching framework for connecting curricular entities is shown in Figure 2. There are four layers represented: the CBK for participating disciplines, the CBK for information assurance, higher order skills that graduates will develop as their education progresses (e.g., understanding the security implications of given combinations of software), and the accepted set of job skills as specified in the CNSS documents. The top layer could also contain the skills and

knowledge needed for graduates to move on to postgraduate degrees and engage in research in computer security.

While the notion of layering knowledge implies a strictly hierarchical relationship between the layers, clearly some of the program outcomes rest directly on the CBK for information assurance or even prior knowledge brought in from the supporting disciplines. For example, CNSS 4011 requires that student have a familiarity with basic computer architecture concepts that would most likely be taught in a Computer Science or Computer Engineering course.

The concentric circles surrounding the four layers articulate the relationship of the curriculum framework within the broader context and with other significant and relevant educational programs and initiatives, 2) enumerate important knowledge, and 3) develop scope and suggested sequences. The top layer of figure 2 suggests that the curricula should be aligned with needed job knowledge and skills. The job knowledge and skills needed in any field are influenced by societal goals and educational aims. Societal goals are what citizens and/or policymakers want the country's political, economic, and social institutions to accomplish; while educational aims are what citizens or policymakers want society's educational institutions (formal and informal) to accomplish to contribute to the achievement of societal goals (6). Educational aims are generally long-term objectives and are the result of many influences including formal education, community education, socialization, maturation, and so on. Educational aims change over time to reflect changes in societal values and serve as the justification for educational goals. Educational goals are what citizens or policymakers want formal educational institutions to accomplish; educational goals reflect what schools/colleges are to accomplish in a broad sense. Educational goals reflect the broad characteristics that are supposed to result from learning over years and across subject matter areas. Educational goals also serve as the justification for learning objectives. Learning objectives are what people are intended to learn as a consequence of being students in educational institutions. Learning objectives refer to intended educational consequences of particular courses, units of study, or even specific lessons. Societal goals, educational aims, educational goals, and learning objectives should be aligned in a purposeful and intentional manner; a principle that is and will continue to guide the development of our work.

While it is possible to infer sequence from the framework, we want to note that how the IA curriculum is approached for instruction, i.e., in a bottom-up, top-down, or project-based manner, is an institutional decision. The hierarchical relationship suggested in figure 2 is not meant to suggest that the material *should* be taught in a classic bottom-up fashion. In fact, we recognize that one of the most powerful paradigms for teaching computer security concepts is to embed appropriate topics in the context of a problem domain. For example, buffer overflow attacks (which account for the majority of network attacks) are easy to understand when added on to a discussion about stack frames for high-level languages. When buffer overflows are studied is isolation in a security course, the discussion is more abstract. Similarly, the implementation of access control and reference monitors fits well with a study of the implementation of file systems in an operating systems course. It is our hope that the model proposed here when instantiated with skills and knowledge will help uncover opportunities to connect course content.

Each of the layers is described in the following sections. Once instantiated, we map backwards from the outcome down through the layers. The relation used is "needs to learn". For example, in order to determine the fitness of a particular password scheme (which would be a higher order skill), we may need to understand how the password is stored and which cryptographic algorithm is used. In order to understand the strength of the cryptographic algorithm, we may need to understand basic number theory principles and algorithmic complexity.

One interesting result when viewing the curriculum in this way is that we can identify outcomes that are not well supported by the curriculum. Additionally, we can easily identify taught material that does not directly support an outcome. The latter is not always undesirable, but this process at least affords the opportunity to make an informed decision on the role of the topic in the curriculum.

In the following sections, we will briefly describe the types of information in each layer.

Layer 1: Prerequisite Body of Knowledge

Information Assurance is a broad multidisciplinary field, drawing on knowledge from Computer Science, Computer Engineering, Mathematics, MIS, Political Science, Law, and many more. For this project, we chose to focus on students with a computer science and engineering background preparing to study computer security in a graduate program. As such, the information assurance topics rest squarely on the CS and CprE curricula, although they may use selected topics from other disciplines.

The supporting disciplines of Computer Science and Computer Engineering each have an identified body of knowledge. Other disciplines, such as Information Technology, are under development. These are the topics thought to be essential for students study in their respective fields. The most mature project of this type is the Computing Curricula 2001. CC2001 defines 14 content areas, each containing several sub areas (see Appendix A). Sub areas are assigned the amount of time needed to cover the material, which provides an indication of the relative importance of the topic (*NB*: the recommended amount of time is not shown in the example in Appendix A).

Layer 2: Information Assurance Body of Knowledge

The information assurance body of knowledge is comprised of disciplinary knowledge and skills from layer 1 as applied to the practice and advancement of information assurance needs, issues, and organizations. The information assurance body of knowledge is informed by all three levels of the curriculum framework and should be aligned to the other layers in a logical, coherent, and systematic manner. The information assurance body of knowledge is technical know how and expertise that extends beyond what a typical computer science/computer engineer/information technology professional would need/be expected to know. For example, all computer science students might be expected to know operating system principles, concurrency, memory management, and so on (7). This would be considered a part of the layer 1 computer science core body of knowledge. The information assurance layer 2 skills that build on the computer

science operating system knowledge might include defining secure operating systems, securing an operating system, and configuring and managing security tools.

Layer 3: Higher Order Skills

The higher order skills layer depicted in figure two represents the skills and abilities that cut across the layer 1 and layer 2 topic areas. Regardless of the disciplinary foundation and the articulation of that foundation to advanced technical IA knowledge, all IA professionals need higher order information assurance skills in the areas of risk assessment, modeling and mitigation; evaluation of the efficacy of competing security mechanisms, methodologies, and models, security requirements, standards, and legal implications and laws.

Layer 4: Job/Professional Level

The fourth and last layer at which we are considering information assurance knowledge and skills is at the job/profession level. Job knowledge and skills are those abilities that graduates need to be specific in professional practice. This includes, but is not limited to, 1) job analyses provided by the Committee on National Systems Security (8), 2) skills recognized by given professional organizations for credentialing, e.g., the common body of knowledge for the Certified Information Systems Security Professional (CISSP) credential (9), and 3) skills needed in research and development.

4. Example

The following intrusion detection example represents how we envision the alignment of layers 1-4. At the layer 1 level, specific intrusion detection job skills expected out of professional might include perform a traffic analysis or monitor systems for accuracy and abnormalities (7). At the layer 2 level, higher order skills required to perform this job task might include understanding measurement basics include validity and reliability of data and/or system composability and how a system can actually be made vulnerable by installing intrusion detection hard/software. At the layer 3 level, intrusion detection spans the management, monitoring auditing and forensics under the area of network security. Finally, the third through first layers are founded on computer science knowledge, e.g., units in log files and pattern matching that might be covered in a junior level class in algorithms and data structures.

5. Current Status

To date two workshops have been held; the outcome of which is a description of the general topics (equivalent to the "areas" level in the ACM/IEEE 2001 Computing Curricula). An example has been provided in Appendix B.

The next step is for the general topics list to be reviewed by a broader community of information assurance educators to determine if the "areas" list in its current state is sufficient, and if not, changes needed. The next step in this project is to then flesh out body of knowledge with sub areas (equivalent to the "units" level in the ACM/IEEE 2001 Computing Curricula) and the relative importance of each unit as denoted by time. While we have a straw man draft of units that belong under each area, this has not been scrutinized by the broader IA community for

completeness, organization, and so on. The next and last step that we hope to take on this project is to complete the second step in the backwards design model, i.e., define performance metrics with the goal of providing the field with a means to demonstrate that graduates have the necessary IA knowledge and skills.

Throughout the process, we noted a number of meta-curricular issues that were documented as follows. Several terms have multiple meaning, e.g., threat, vulnerability, validation, verification, testing, secret key, certificate, one-way functions, social engineering, risk, security, proof, policy, security tools, undergraduate, graduate, curriculum (and more to come). Care should be taken to operationally define these terms so that others (including students) can better understand their multiple meanings in context. Throughout the undergraduate curriculum we should also discuss existing tools and resources such as BugTraq, and CERT Advisories, to name a few. Depending upon the students' interests, undergraduate programs might also want to discuss open research issues. Students should be required to write large programs, maintain programs overtime, and work in teams. Students are not trained to be professional programmers working in teams on large codes. This is perceived as a source of many security problems. IA education encompasses the issues that arose from the military defense world and has grown to include e-commerce, e-government, e-learning (and others) and students need to understand this evolution and spectrum. Students need to understand the notion of "no such thing as absolutely secure".

There are also personal characteristics associated with being an IA professional that students should understand so they can self-assess whether or not they will be satisfied with a career in IA. Such characteristics include: detail-oriented, high level of self-discipline, voluntary paranoia. To address how to integrate detail-orientation into the undergraduate curriculum, we can look at other disciplines where attention to detail is also paramount. Finally, at the undergraduate level, it was assumed that students graduating from programs that include these topics are expected to go into the following types of careers: Low Level IT Engineer, System Administrator with a Security Specialization, Programmer with a Security Specialization, Network Engineer with Security Specialization, or a Security Software Developer. It was also assumed that students would have taken more than one 4th generation language course so that students have the ability to program.

6. References

(1) The Steelman draft (August 1, 2001) is at http://computer.org/education/cc2001/steelman/cc2001/index.htm
(2) National Security Telecommunications and Information Systems Security Instructions. http://www.nstissc.gov/html/library.html
(3) Information Technology Association of America survey, May 2002, http://www.itaa.org/news/pr/PressRelease.cfm?ReleaseID=1020695700
(4) www.house.gov/science/hearing_105.htm#Technology
(5) Wiggins, G., and McTighe, J. (1998) Understanding by Design. Association for Supervision and Curriculum Development; Alexandria, VA.
(6) Posner, G. (1992). Analyzing the Curriculum. McGraw Hill; New York.
(7) ACM/IEEE 2001 Computing Curricula. http://www.computer.org/education/cc2001/final/index.htm

(8) NSTISSI. National Training Standard for Information System Security Professionals 4011 http://www.nstissc.gov/html/library.html
(9) ISC^2 Common Body of Knowledge http://www.isc2.org/cgi/content.cgi?category=8

Appendix A. ACM/IEEE 2001 Computing Curricula Main Areas and Sample Units

The CS body of knowledge is organized hierarchically into three levels. The highest level of the hierarchy is the **area,** which represents a particular disciplinary subfield. Each area is identified by a two-letter abbreviation, such as OS for *operating systems* or PL for *programming languages*. The areas are broken down into smaller divisions called **units,** which represent individual thematic modules within an area. Each unit is further subdivided into a set of **topics,** which are the lowest level of the hierarchy (not shown here).

Areas
Discrete Structures (DS)
Programming Fundamentals (PF)
Algorithms and Complexity (AL)
Architecture and Organization (AR)
Operating Systems (OS)
Net-Centric Computing (NC)
Programming Languages (PL)
Human-Computer Interaction (HC)
Graphics and Visual Computing (GV)
Intelligent Systems (IS)
Information Management (IM)
Social and Professional Issues (SP)
Software Engineering (SE)
Computational Science and Numerical Methods (CN)

Sample Units in Programming Fundamentals Area

PF1. Fundamental programming constructs [core]
PF2. Algorithms and problem-solving [core]
PF3. Fundamental data structures [core]
PF4. Recursion [core]
PF5. Event-driven programming

Appendix B: Topics in Four Focus Areas

Topics in Cryptography

The development of cryptography
- First principles
  - Protecting confidentiality
  - Ensuring integrity
  - Guaranteeing authenticity
- Historical cryptography
  - Substitution ciphers
  - Transposition
  - Frequency-based cryptanalysis
  - Codes & Code machines

Fundamentals
- Block vs stream ciphers
- Chaining
- Threshold cryptography
- Zero-knowledge proofs
- Oblivious transfer
- Pseudo-random number generators
- Secret sharing
- Key management and key distribution
- Key space

Important symmetric algorithms
- DES
- AES
- Clipper / Skipjack
- RCn

Asymmetric algorithms
- Public key cryptography
- RSA
- Elliptic curve cryptosystem
- Digital Signature Algorithm

Cryptographic protocols
- Identification, authentication and authorization
- Role of encryption
- Frameworks for secure e-commerce
- Third-party certification authorities
- Single sign-on
- Electronic voting
- Electronic contracts & non-repudiation

Hardware implementations
- Cost/benefit analysis
- Enforcement
- Digital rights
- Vulnerabilities
- Crypto processors

Digital signatures
- Definitions & Benefits
- Mechanisms
- Certificates

Applications of cryptography
- Cryptography in the OSI model
  - IPv6
- IPSec
- Smartcards
- Biometrics

Public key infrastructure and certificate authorities
- Need for public key cryptosystem
- Need for public key infrastructure
- Public key certificate
- Key revocation
- Key recovery

Implementation issues
- Algorithmic weakness vs implementation weakness
- Secrecy of the algorithm is not a defense
- Types of attacks
- Overview of non-brute-force attacks
- Product certifications
  - Common Criteria
  - Commercial standards
- Key escrow

Cryptanalysis
- Strategies
  - Brute-force
  - Linear and differential cryptanalysis
  - Meet-in-the-middle/birthday attack
  - Timing analysis
  - Side-channel analysis
- Analysis of randomness
- Interception techniques
- Reverse engineering
- Hardware failures

Steganography
- Examples
- Analysis
- Defenses

Latest developments
- Chaffing and winnowing
- Recent algorithms
- New products
- Quantum computing effects on cryptanalysis
- Quantum cryptography

Topics in Secure Computing Systems

Access control
- ACLs
- capabilities
- Data- and user-oriented access control
- multi-level security
- Simultaneous access

Identification, authentication and authorization
- accounting
- authentication
- authorization
- biometrics
- identification
- passwords
- tokens

Design of secure systems
- architectural implications of OS for security
- design principles
- hardening OSs
- high-availability / sustainability
- inference control
- Protection based on an operating system mode
- Protection of memory
- reference monitor
- security kernels
- survival
- system design principles
- trusted operating systems; e.g., trusted LINUX
- malicious software: analysis, prevention

Evaluation
- Common Criteria
- covert channels
- evaluation of secure systems
- penetration testing
- virus prevention

Databases and applications
- application security -- Web servers
- database security
- developing secure distributed applications (JAVA etc.)
- secure file systems
- security databases (active directory, RADIUS, token servers, Kerberos…)

Software development
- authenticating libraries, DLL, run-time
- buffer overflows
- develop security tools (e.g., IDS, sniffer, integrity check)
- how to write secure software
- open-source vs proprietary software and security
- quality assurance and security
- software security
- writing code
- writing patches

Auditing
- application logging
- computer forensics/auditing and system logs, utilities, data
- known vulnerabilities
- logging
- intrusion detection

Operations management
- patching systems
- physical security
- version control

Topics in Network Security

Protocols	**Application-layer services**
IPSec	DNS Domain Name System
IPv6	E-commerce payment systems
key management protocols	e-mail
multicast security	NAT
raw sockets	SMTP
routing authentication	Web
routing protocols	**Management, monitoring, auditing & forensics**
SSH	management
TCP / UDP	SNMP
TCP state analysis	honeypots
tunneling	intrusion detection
VPN	monitoring
Network basics	network forensics
ISO/OSI model	traceback
Network design	**Infrastructure**
topology	dialup security
transport-level security	Ethernet switching (VLANs, . . .)
Vulnerabilities	grid security
NOS weaknesses	media
protocol vulnerabilities	middleware
sequence-number prediction	PKI
vulnerabilities at the different layers of the OSI	protection of network infrastructure (e.g., secure routing protocols)
Attacks	RFI radio frequency interference
DoS	TEMPEST / emanations control
eavesdropping	WANs
man-in-the-middle attacks	**Wireless & broadband**
sniffing	Bluetooth
spoofing	broadband
steganography	DSL
types of attacks (exploitation of protocol weaknesses)	satellite
	Cable
	GB Ethernet security
	WEP
	Filtering
	filtering mechanisms: static, stateful, proxy, . .
	firewalls

Management, Policy and Response

- **Security policy guidelines**
  - Terminology
  - Resources for policy writers
  - Writing the policies
  - Organizing the policies
  - Presenting the policies
  - Maintaining policies
- **Security awareness**
- **Ethical decision-making and high technology**
- **Employment practices and policies**
  - Hiring
  - Management
  - Termination of employment
- **Operations security and production controls**
  - Basic concepts
  - Operations management
  - Providing a trusted operating system
  - Protection of data
  - Data validation
- **E-mail and Internet use policies**
- **Social psychology to implement security policies**
- **Auditing and assessing computer systems**
- **Cyberspace law and computer forensics**
  - Contracts
  - Defamation
  - Due diligence and private liability
  - Indecency and obscenity
  - Litigation
  - Criminal acts
  - Investigation
- **Privacy in cyberspace**
  - Worldwide trends
  - European approaches to privacy
  - United states
  - Compliance models
- **Protecting intellectual property**
- **Security standards for products**
  - Security assessment standards associated with security implementations
  - Establishing trust in products and systems and managing risks
  - Common criteria paradigm
- **Management responsibilities and liabilities**
  - Responsibilities
  - Liabilities
  - Computer management functions
  - Security administration
- **Developing security policies**
- **Risk assessment and risk management**
- **Incident Response and Recovery**
  - Computer emergency quick-response teams
  - Data backup and recovery
  - Business continuity planning
  - Disaster recovery
  - Insurance relief
  - Working with law enforcement
    - Goals of law enforcement
    - History of law enforcement and computer crime
    - Anatomy of a criminal investigation
    - Establishing relationships with law enforcement
  - Developing internal investigative capabilities
    - Internal investigations
    - International investigations
    - Computer evidence
    - Decision to report computer crime