Towards Information Assurance (IA) Curricular Guidelines

Stephen Cooper, Leader
Stanford University
353 Serra Mall
Stanford, CA 94305
1.650.723.2273
coopers@stanford.edu

Christine Nickell, Co-Leader
Department of Defense
9800 Savage Road Suite 6722
Ft. Meade, MD 20755
1.410.854.6206
c.nicke2@radium.ncsc.mil

Lance C. Pérez, Co-Leader
University of Nebraska
209N SEC
Lincoln, NE 68588
1.402.472.6258
lperez@unl.edu

Brenda Oldfield, Co-Leader
Department of Homeland Security
NCSD
Washington DC 20528
1.703.235.5184
Brenda.Oldfield@dhs.gov

Joel Brynielsson
Royal Institute of Technology
SE-100 44 Stockholm
Sweden
46.70.5394300
joel@kth.se

Asim Gencer Gökce
TÜBİTAK / UEKAЕ
Tunali Hilmi Cad. Binnaz Sk.
No:2 Kat:3 06700 Kavaklidere
Ankara, Turkey
90.312.4277366 x302
gencer@uekae.tubitak.gov.tr

Elizabeth K. Hawthorne
Union County College
1033 Springfield Ave
Cranford, NJ 07016
1.908.497.4232
hawthorne@ucc.edu

Karl J. Klee
Alfred State College
10 Upper College Drive
Alfred, NY 14802
1.607.423.2665
kleekj@alfredstate.edu

Andrea Lawrence
Spelman College
350 Spelman Lane
Atlanta, GA 30314
1.404.681.3643
Lawrence@spelman.edu

Susanne Wetzel
Stevens Institute of Technology
Castle Point on Hudson
Hoboken, NJ 07030
1.201.216.5610
swetzel@stevens.edu

ABSTRACT
Information assurance and information security are serious worldwide concerns. Computer security is one of the three new focal areas of the ACM/IEEE's Computer Science Curriculum update in 2008. This ACM/IEEE report describes, as the first of its three recent trends, "the emergence of security as a major area of concern." [3]

The purpose of this working group report is to continue the work of the 2009 working group on information assurance (IA) education. The focus of the 2010 working group is to examine the curricula of existing academic programs, as well as at the key academic governmental and industry IA education standards and guidelines identified by the 2009 IA working group in order to begin defining the IA education space as a first step towards developing curricular guidelines.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Categories and Subject Descriptors
K.3.2 Computer and Information Science Education

General Terms
Security.

Keywords
Information Assurance, IA, Education, Standards, Guidelines.

1. INTRODUCTION
Information assurance (IA) is a set of technical and managerial controls designed to ensure the confidentiality, possession of control, integrity, authenticity, availability, and utility of information and information systems [11]. IA includes measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. These measures include providing for restoration of information systems by incorporating protection, detection, and reaction capabilities.

Information assurance is of both national and international significance because of the increased reliance of governmental, military, public infrastructure and financial functions on complex...
interconnected computer systems and networks. These systems not only store information, they exchange and process information and are involved in increasingly significant decision processes that demand all aspects of information assurance. With this increased reliance on electronic infrastructure, however, has come the realization that these systems are vulnerable to a myriad of attacks, many of them cyber in nature and not requiring the resources of a world power. This combination of a desire to continue to gain the benefits of complex electronics systems with the recognition of their inherent vulnerabilities has made IA a global priority.

In the Cyberspace Policy Review [13], the report to U.S. President Obama, Melissa Hathaway, who was serving as Acting Senior Director for Cyberspace for the National Security and Homeland Security Councils, presented a call to arms to address the nation's vulnerabilities with respect to information assurance. Two of the most important observations in this report are that the U.S. needs to produce far more IT graduates (for both the public and private sectors) in general, and students with IA capabilities in particular. The second observation is that the U.S. cannot address the challenges of cybersecurity alone. Currently, no country is producing enough IA graduates and, perhaps even more alarmingly, no country has the capacity to quickly produce the numbers of qualified IA graduates that are needed now or in the near future. For example, in the U.S., there are now 117 universities that have received designation from the National Security Agency as Centers for Academic Excellence (CAEs) in Information Assurance Education (CAE/IAE) or in Information Assurance Research (CAE-R) [14]. In 2010, for the first time, six community colleges received the Center for Academic Excellence in Information Assurance two-year Education (CAE2Y) designation, serving as national IA education and training models [12]. These CAEs represent approximately 3% of the 3,000 plus undergraduate institutions in the U.S. And the numbers are not much better when looking at the number of IA programs (as a percentage of the total number of universities) in Australia, the UK, and the rest of the world.

Last year during the 2009 Conference on Innovation and Technology in Computer Science Education (ITiCSE), the "Addressing Information Assurance Education Standards" working group undertook a retrospective study of the state of IA education. The working group consisted of 15 faculty, researchers, and government officials from Australia, Sweden, the UK, and the US with a broad range of experience in IA in the educational, industrial, and government sectors. The working group's paper [6] presented a brief history of IA education followed by a comprehensive review of extant academic, government, and industry standards and guidelines in the field. The paper also considered the issues of maintenance and updating of standards and the assessment of IA training and educational programs. Through its exploration of existing IA standards and guidelines, as well as in its discovery of what guidance is being provided from other areas of computing, the paper established a foundation on which future efforts to develop IA curricular guidelines could build. A key conclusion of the 2009 working group was that it was necessary to have a concerted global effort to develop models for IA education appropriate for two-year, four-year and graduate degrees housed in a variety of academic units and that these models should rely less on training standards and more on modern pedagogical and educational practices.

This year, the "Towards Information Assurance Curricular Guidelines" working group endeavored to build on this foundation and to move forward with the first steps toward defining IA educational models. To this end, the working group attacked two specific problems. First, the working group sought to identify the set of topic areas that comprehensively define the field of IA, independent of the type of degree program and the specific discipline of the academic unit in which it resides. Second, the working group sought to develop a set of topics and associated student-learning outcomes for one particular subject in IA. This set of topics and corresponding student-learning outcomes can hopefully serve as a model for future efforts to similarly define the other subjects.

The remainder of this paper is divided into two main sections. In section 2, the space of what IA education encompasses is explored and a proposed Body of Knowledge (BoK) for IA education is described. In section 3, a single IA subject, secure coding, contained in the proposed Body of Knowledge, is examined in detail and a set of student learning outcomes developed. Secure coding was selected because it is a significant source of vulnerabilities and has become the subject of several high-level IA discussions. In addition, secure coding is a subject whose coverage and competencies can vary greatly depending on the focus of the particular IA program in which it is taught and was thus judged to be a challenging subject from a curricular perspective that would provide a better example of the feasibility of developing a single curricular model. Finally, the paper identifies some challenges and future directions.

2. A BODY OF KNOWLEDGE FOR IA EDUCATION

2.1 CAE Survey and Results

Before the ITiCSE Information Assurance Curriculum Guidelines Working group convened at Bilkent, an electronic survey was distributed to the 117 existing Centers of Excellence and the six community colleges that recently became CAEs. Although this was not a random sample, it was sent to the institutions known to have a CAE and thus a program or degree in IA. It needs to be emphasized that the IA programs at the CAE's take on many forms, existing in variety of types of academic units, and have several different specializations. The primary goal of the survey was to synthesize the varied curricular content of these programs into a single comprehensive set of topic areas that would comprehensively define the space of IA education. Individual programs may teach different subsets of these topic areas and with differing emphasis.

Thirty-three responses were received (by June 26, 2010, when this working group commenced its efforts) of which 29 were usable for the data analysis phase. The questions included demographics of the institutions and names of the degree programs where IA programs are housed. The question was also raised as to whether the program was an associate’s degree, a bachelor’s degree, a master’s degree, a certificate, or other.

1. What school/organization are you from?
2. What program(s) are offered by your institution in information assurance?
3. What disciplines are most closely associated with information assurance at your institution? Check all that apply.

Two additional questions were posed:

4. In your current program, what percentage of the course of study covers each of the following major topic areas?

5. In an ideal information assurance program, what percentage of the course of study should cover each of the following major topic areas?

The topic areas listed in the survey were: Fundamental Aspects, Cryptography, Ethics, Policy, Digital Forensics, Access Controls, Security Architecture, Network Security, Risk Management, Attacks/Defenses, Operational Issues, and Secure Software Design and Engineering. Space was also allowed for the respondents to add topic areas they deemed to be missing from the suggested list. In the survey, sample subjects were provided for some of the topic areas in order to further clarify them. These suggested subjects were not comprehensive and respondents were not asked to evaluate them.

The analysis of the questions indicates the following. For the second question, of the 29 institutions, 11 (37.93%) of them offer a Certificate Program, 5 (17.24%) of them offer an Associate’s Degree, 16 (55.17%) of them offer a Baccalaureate Degree, 19 (65.52%) of them offer a Master’s Degree, and 11 (37.93%) of them offer a Doctoral Degree.

Further data analysis of questions four and five yields the following insights: The topics listed were regarded as sufficient to cover the field by the respondents as none of the respondents indicated any omissions, which could have been entered under the option “other”. (One respondent entered identity management, but the working group considers this to be a subject contained within the access control area.) For both questions, the respondents were also asked to provide the percentage of a specific subject being covered as part of the IA program. For some respondents, the sum of their responses added up to more than 100% and, consequently, these responses were normalized to 100%. Subsequently, the averages were computed over all responses. Table 1 provides the details for all subjects and both questions four and five.

The findings of the survey were then used by the working group as a basis from which to define the areas of an all-encompassing IA body of knowledge (BoK). Each area was then further refined into a set of subjects. This naming convention resembles that of Computing Curricula 2001 [4], but differs in content with regard to the way learning outcomes are formulated. In the following subsections, the details are given of the working groups' proposed IA area and subject taxonomy. A table summarizing this taxonomy may be found in Appendix A.

Additional efforts will be necessary to allocate core hours to each area and determine what electives should be part of IA programs housed within CS, IT, Business, Security, Engineering, or Technology. Furthermore, future work should explore what security areas and subject should be part of the general degree programs in CS, IT, Business, Security, Engineering, and Technology.

### 2.2 The Working Group’s Proposed Body of Knowledge

#### 2.2.1 Changes in Areas and Suggested Subjects

After considerable deliberation, the working group decided to remove *Operational Issues* from the list of topic areas. In the survey, the only subject suggested within Operational Issues was "practicum". The working group felt that hands-on labs, teaching, and exercises are essential to the entire IA curriculum and should be integrated into each area and not be considered a single distinct area. Furthermore, operational issues as defined, e.g., as part of CISSP [8], are already included and represented in the other areas. Consequently, *Operational Issues* was removed as a stand-alone area. Also, *Fundamental Aspects* was change to *Fundamental Concepts* to reflect the more learning centered philosophy of the curriculum.
In some cases, subjects that were suggested under specific areas in the original survey were moved to other, more appropriate areas by the working group. For example, *Theory of Security Policies* that was suggested under Access Control was moved to the topic area of *Security Policies*. More importantly, the suggested topics in the original survey were greatly extended with the intention to provide a comprehensive listing for each area. Furthermore, in expanding the list of subjects the working group tried to keep the subjects relatively general in order to provide some abstraction from specific details such as access control lists or backup.

### 2.2.2 Details on Areas and Subjects

The IA body of knowledge proposed by the working group is now given in detail. As noted earlier, the BoK has been organized hierarchically into fields, denoted areas, which are further subdivided into more specific subjects in order to resemble the Computing Curriculum 2001.

**Area: Fundamental Concepts**

This area includes the fundamental concepts and basic definitions needed to understand the field as well as an overview of the field as preparation for future study.

**Subjects:**

- *History* of the discipline of Information Assurance.
- *Basic Terminology* that should be recognized by those studying the field.
- *Information Assurance Concepts* that are key to building an understanding of the IA area.
- *Awareness* of the areas of Information Assurance and important aspects of these areas.
- *National and Cultural Differences* including topics such as HIPAA, Safe Harbor, and data protection laws.

**Area: Cryptography**

Cryptography covers the basic theoretical concepts, mechanisms, algorithms and protocols. It includes both the design as well as the analysis of the various primitives and protocols.

**Subjects:**

- *Applications of Cryptography* motivate the importance of cryptography. The applications range from networking, wireless protocols, to day-to-day applications such as secure email, credit cards, car keys, garage door openers, and car engines, to critical infrastructure elements such as electrical grid systems.
- The *Mathematical Preliminaries* include topics in linear algebra, number theory, probability theory, and statistics.
- *Cryptography Legal Issues* include discussions on export control matters, key escrow with a focus on national and international regulations.
- The *Basic Cryptography Terminology* covers notions pertaining to the different (communication) partners, secure/unsecure channel, attackers and their capabilities, encryption, decryption, keys and their characteristics, signatures, etc.
- The subject of *Formal Definitions* introduces notions such as indistinguishability, one-way permutations, one-way functions, pseudo-randomness, etc.
- The *Historical Methods* include ciphers such as the shift cipher, Caesar cipher, affine cipher, etc., together with typical attack methods such as frequency analysis, etc.
- The center role of *Information Theory* to cryptography starts with the work by Shannon and ranges to current treatments of information theoretic security analysis for state-of-the-art primitives and protocols.
- *Cryptographic Primitives* include encryption (stream ciphers, block ciphers public key encryption), digital signatures, message authentication codes, and hash functions.
- *Cryptanalysis* covers the state-of-the-art methods including differential cryptanalysis, linear cryptanalysis, factoring, solving discrete logarithm problem, lattice-based methods, etc.
- *Cryptographic Algorithm Design* covers principles that govern the design of the various cryptographic primitives, especially block ciphers and hash functions.
- The treatment of *Common Protocols* includes (but should not be limited to) current protocols such as RSA, DES, DSA, AES, ElGamal, MD5, SHA-1, Diffie-Hellman Key exchange, identification and authentication protocols, secret sharing, multi-party computation, etc.
- *Public Key Infrastructure* deals with challenges, opportunities, local infrastructures, and national infrastructure.

**Area: Security Ethics**

This area covers the ethical issues involved in information assurance with emphasis on the aspects of ethics that deal with security.

**Subjects:**

- *Privacy Issues* and how security may be used to promote and protect privacy.
- *Hacking and Cracking* viewed from two perspectives: ethical issues of hacking and cracking.
- *Legal Issues* involved in security breaches and misuse of access.
- Prevalent *Ethical Dilemmas* in IA, such as those involving data or system access, such as misuse of data obtained or whistle blowing, with case studies where possible.
- *National/Cultural Differences* affect what is ethically and legally correct.
- *Bases for ethical decision making* such as philosophy, codes of ethics/professionalism, moral judgment models, etc.
- Challenges in balancing freedom of information and security.
- Security as a societal goal including the relation of security to other socially desirable goals, such as liberty, justice, economic well being, etc.
- Legal vs. ethical aspects.

**Area: Security Policy**
A security policy consists of objectives, goals, rules and formal procedures that form the foundation and structure for developing a comprehensive security program within an organization. Policy compliance languages are used to enforce policy compliance and legal restrictions within an organization. There are also organizational and governmental security policies that dictate secure usage of personal and other sensitive information.

**Subjects:**
- Strategies and Plans for creating security policies.
- Policies, Guidelines, Standards and Best Practices for individuals or organizations, including national security policies.
- Procedures for creating policies, guidelines, standards, specifications, regulations and laws.
- Privacy Policies to help protect personal and other sensitive information.
- Compliance and Enforcement of policies, standards, regulations, and laws.
- Formal Policy Models such as Bell-LaPadula, Biba and Clark-Wilson, which provide precise specifications of security objectives.
- Policy Languages provide a means to express policies.
- Relation of national security policies, regulations, organizational security policies, formal policy models, and policy languages.
- Notions of Security include unconditional security, complexity-theoretic security, computational security, provable security, as well as the treatment of the current models for security analysis including the random oracle model and the standard model.

**Area: Digital Forensics**
Digital Forensics deals with collecting, preserving, analyzing, and reconstructing stored data evidence, including volatile data, from a crime scene where a computer was involved.

**Subjects:**
- Basic Principles and Methodologies for digital forensics.
- Computer/network/system attacks.
- Rules of Evidence – general concepts and differences between jurisdictions.
- Search and Seizure of evidence, e.g., computers, including search warrant issues.
- Digital Evidence methods and standards.
- Data analysis and validation.
- Techniques and standards for Preservation of Data.
- Legal and Reporting Issues including working as an expert witness.

**Area: Access Control**
Access control is the collection of mechanisms that enable an authority to control access to resources in an information system. Access control allows specification of what users of the information system can do, which resources they can access, and what operations they can perform on the information system.

**Subjects:**
- Basic Principles of an access control system prevent unauthorized access.
- Physical Access Control determines who is allowed to enter or exit, where the user is allowed to enter or exit, and when the user is allowed to enter or exit.
- Technical/System Access Control is the process of preventing unauthorized users or services to utilize information systems.

**Area: Security Architecture and Systems**
Security Architecture and Systems deals with the various aspects arising in architecting secure complex systems. This includes technical aspects such as architecting hardware and software as well as overarching systems including multiple components and users.

**Subjects:**
- How to secure Hardware, including how to make hardware tokens and chip cards tamper-proof and tamper-resistance.
- Implementation issues.
- Usability includes the difficulty for humans to deal with security (e.g., remembering PINs), social engineering, phishing, and other similar attacks.
- Analyzing and identifying System Threats and Vulnerabilities.
- Investigating Operating Systems Security for various systems.
- Multi-level/Multi-lateral Security.
- Design and Testing for architectures and systems of different scale.
- Penetration Testing in the system setting.
- Products available in the marketplace.

**Area: Network Security**
Today’s network infrastructures are frequently exploited either as the target of attacks or as the medium that enables to attack on end systems. Network security is to protect data during transmission, resources from disclosure and systems from network-based attacks.

**Subjects:**
- Network Basics are the prerequisites of network security that cover the fundamental concepts of networking.
- Protocols are the commonly used network communication protocols including TCP/IP, WAN, LAN technologies.
- The network Standards are important for facilitating the interoperability of networked systems since all of the network devices and technologies have their own internal interfaces.
- Network Vulnerabilities are present in every networked system today. There are external network vulnerabilities that mainly result from coordinated attacks or data
interception and internal network vulnerabilities that are mainly caused by overuse of network resources.

- Different types of Defense Mechanisms / Countermeasures enable network administrators to secure their network systems.
- Network Design is an important step in securing a network.
- The importance of continuity, reliability, and security of a networked system makes Network Auditing an essential subject.
- In order to secure a network, the security of the network should be evaluated by internal and external Penetration Testing.

Area: Risk Management

A risk in a computer environment depends upon the level of the threat or vulnerability and the value of the information asset. Risk management is the total process of identifying, measuring, and minimizing uncertain events affecting resources. Risk management techniques look for ways to balance the cost of protection versus the value of the asset.

Subjects:

- Risk Analysis involves identifying the assets, probable threats, vulnerabilities and control measures to discern risk levels and likelihoods. It can be applied to a program, organization, sector, etc. Knowledge in this area includes knowing different risk analysis models and methods, their strengths and benefits and the appropriateness of the different methods and models given the situation.
- Cost/Benefit Analysis is used to weigh private and/or public costs versus benefits and can be applied to security policies, investments, programs, tools, deployments, etc.
- Continuity Planning will help organizations deliver critical services and ensure survival.
- Defense In Depth strategies constitute the process of layering defenses to provide added protection that increases security by raising the cost of an attack.
- Disaster Recovery will help an organization continue normal operations in a minimum amount of time with a minimum amount of disruption and cost.
- A Security Audit is a systematic assessment of an organization’s system measuring the conformity vis-à-vis a set of pre-established criteria.
- Asset Management minimizes the life cost of assets and includes critical factors such as risk or business continuity.
- Risk management utilizes the results of risk analysis to develop a systematic plan for managing the risk to include periodic reassessment of risk levels.
- Risk communication
- Enforcement of risk management policies is critical for an organization.

Area: Attack/Defense

A distinguishing feature of IA compared to many other disciplines is that opposing actors need to be taken into account. Security is essentially a human condition and therefore, IA graduates must think about the human-computer interface carefully. This includes an “opponent perspective” that spans both potentially willful, motivated and skilled opponents as well as how uninformed users contribute to insecurity. The area of attack/defense takes the “opponent perspective” into account by considering technology and processes using an attack versus defend viewpoint.

Aside from the content in the section of attack/defense, the attack/defend section refers to a “mentality”. Security is constantly moving and students need exposure to this frame of mind. Just as graduates need to understand that attack/defend is a frame of mind, students need an appreciation that security is temporal and dynamic. It is a relative state and is relative to a number of factors including the past, the changing nature of threats, vulnerabilities, relative to culture, relative to a mindset, relative to other priorities (be they freedom of information, profit, privacy, economic well-being, a checklist of best practices). This relativity should not be underestimated.

Subjects:

- Threats and Vulnerabilities focus on being aware of the possible threats vis-à-vis the possible vulnerabilities that may be targeted.
- Types of Attacks depend on a multitude of factors, e.g., the protocol level being attacked, physical access, social engineering, etc.
- Types of Attackers are important to know about in order to defend oneself, e.g., teenagers attacking for the fun of it, professionals attacking to reveal security holes, criminals attacking for money, states attacking for political reasons, etc.
- Defense Mechanisms may be applied to protect an asset.
- Incident Response includes putting together the resources and applying the appropriate defense mechanisms to challenge an incident.

Area: Secure Software Design and Engineering

Secure software design and engineering deals with producing software that meets specified security requirements. Depending on the requirements, many different techniques and/or processes may be used to meet the required level of security. A higher level of security calls for more advanced techniques/processes, which cost more.

Subjects:

- Secure Software Specification deals with specifying what the program should and should not do, which can be done either using a requirements document or using a more formal mathematical specification.
- Secure Coding involves applying the correct balance of theory and practice to minimize vulnerabilities in code.
- Secure Testing is the process of testing that security requirements are met.
- Program Verification and Simulation is the process of ensuring that a certain version of a certain implementation meets the required security goals, either by a mathematical proof or by simulation.
- **Language-based Security** addresses the variety of security protections and problems associated with different programming languages.
- **Secure Design** is the process of designing software so that it meets the security requirements.
- **Maintenance** deals with maintaining the required security level throughout the lifetime of the software meaning, e.g., those new versions of a piece of software must be re-verified to meet the security requirements.

## 2.3 Theory versus Practice

Compared to regular university courses in such subjects as computer science and mathematics, best practices within IA education are far from obvious. The subject can be tackled using many different viewpoints and perspectives. As an example, apart from computer knowledge, a skilled IA person also needs to be creative enough to be predictive in order to be ahead of the attacker. In this sense, teaching IA is more difficult than teaching, e.g., ordinary computer programming skills, which gives rise to discussions about theory vs. practice, training vs. scholarship, etc.

Among the distinguishing features of IA education, the most significant one compared to other disciplines is that opposing actors need to be taken into account. That is, besides the actual computer skills, real world IA deals with various aspects of conflict, i.e., aspects of conflict that are usually discussed within the social sciences with pure conflict being handled by economists using game-theoretic methodology being the extreme. This “opponent perspective” makes IA education quite different from that of other education where technical skills are taught without considering a potentially willful, motivated and skilled opponent. Hence, IA education needs to additionally be designed from an attack versus defend viewpoint, similar to other educational efforts described by, e.g., [2] or [9].

Related to the attack/defend viewpoint, but somewhat different from ordinary societal crisis management, is that the very same (computer) tools and the very same knowledge can be used for both good and bad purposes. From a pedagogical point of view, it follows that offensive and defensive means are often the same, or at least heavily intertwined: when learning about security one needs to understand both the mindset of the attacker and how to perform the actual exploitation of vulnerabilities. In fact, it would be even easier to educate computer criminals than security-aware professionals; to attack a system it suffices to find a single vulnerability, but to defend a system all vulnerabilities must be found and repaired. Since the attacker and the defender use the same tools and the same knowledge to find the same vulnerabilities, it follows that it is often easier to attack than to defend. As a result, it follows that hands-on IA training may benefit from letting students—in one way or the other—initiate attacks in a controlled way to learn about prevention and self-defense. Focusing on student activities—“what the student does”—to learn real life (technical) IA skills using such a “learning by doing” approach goes hand-in-hand with current pedagogical trends [1].

In conclusion, IA teaching could potentially benefit from a number of practically-oriented teaching methods based on hands-on laboratory experiments that focus on problem solving assignments, the idea being to acquire an insightful understanding of the field as a whole by being able to relate to appropriate tasks that represent real-life situations. At one extreme, this practically oriented approach to learning IA topics has been used for letting so-called cyber defense exercises shape the content of whole curricula [10]. For the purpose of constructing IA teaching guidelines, however, it is noted that hands-on training, cyber defense exercises, and the like are examples of pedagogical methods that can be used to teach more general knowledge. Hence, this working group considers such practical hands-on training as important means that can be used to reach the learning goals, but they do not represent the body of knowledge in itself. In addition to these comments about how to teach attack and defend, inculcating a sense of civic responsibility in IA students is also important. In doing the hands-on/practical exercises, it can be useful to ensure that students do not lose sight of the people they are protecting and how.

## 3. STUDENT LEARNING OUTCOMES

This section develops a Computing Curriculum 2001-like description of a single subject, and identifies the topics and learning outcomes associated with that subject. The purpose is to develop a template that can be used for all of the IA subjects within the eleven areas described above. Throughout the remainder of the document, Computing Curriculum 2001 [4] is referred to as CC2001.

### 3.1 Secure Coding

The subject chosen for exploration is secure coding. Secure coding has attracted a great deal of attention, both within the news (as the absence of secure coding has been blamed for many security failures), as well as at the government-level, where there are, for example, several bills before the U.S. Congress with specific focus on secure coding, and how to ensure that all students who learn to program learn to do so securely.

This is a challenging topic. The schools and universities that currently include a course in secure coding typically do so at an advanced undergraduate or at a graduate level. Because secure coding appears so late in the curriculum, we are in the situation where we are trying to correct bad habits that have been reinforced for several years (“insecure programming”), even though it would seem to make more sense to teach students secure coding habits when they are first taught to program.

### 3.2 The Template

The working group started with the same template that was used for CC2001 [4] – namely the listing for each subject (CC2001 [4] uses the term “unit” to refer to a subject), together with the topics that make up the subject, and the student learning outcomes associated with that subject. However, the working group made several changes to the CC2001 [4] template. The changes and the associated rationales for the changes are summarized below:

1) For each topic, the working group provided a detailed description of that topic, including a definition, list of key principles associated with that topic, and a list of common issues (or examples) associated with that topic. This was necessary to do as each topic can be interpreted quite broadly, due to the interdisciplinary nature of IA.

2) As is commonly part of engineering accreditation (especially ABET), the working group decided to
include an assessment rubric for each student learning outcome to ensure that each outcome is focused and measurable. The working group believes that the creation of similar rubrics would be useful for the CC2001 [4] successor as well.

3) The working group decided to build the student learning outcomes using verbs associated with Bloom’s taxonomy, associating each learning outcome with one of the six levels: knowledge, comprehension, application, analysis, synthesis, or evaluation. In particular, the working group used [5], which categorized various verbs with the above-mentioned six levels in Bloom’s taxonomy. The working group added the verb “code”, which it considered to be at the application level.

4) The use of Bloom’s taxonomy led to finer-grained student learning outcomes (particularly at the lower four levels of the taxonomy), which resulted in a greater number of more specific learning outcomes.

5) The working group decided that rather than making topics core or elective, it made more sense to consider specific learning outcomes as core or elective. Its rationale was that many of the topics do need to be covered, albeit at various depths, depending on the specific academic focus of the particular IA program.

3.2.1 A Challenge in Determining Whether a Learning Outcome should be Core or Elective

As mentioned earlier, IA programs are housed in many different types of departments and the emphasis of any particular IA program varies accordingly. Given that an IA program located within a school of public policy is likely to have a significantly different focus than an IA program housed within a school of computer science, the core topics across all IA programs will be minimal.

3.3 Secure Coding Learning Outcomes

Secure coding is often known as software security, which is probably a more descriptive name for this subject. However, the working group decided to use the term that is more widely used in industry.

3.3.1 Secure Coding Topics, and their Descriptions

There are six topics associated with secure coding:

A. Data Protection

Description: Data protection may be defined as the methods for preventing unauthorized access or manipulation of data, whether internal or external. Key principles include programming, techniques such as information hiding/encapsulation, data flow, strong data typing, and system access controls. Common issues associated with data protection include problems with data corruption, inadequate access controls, poor usage of these controls and canonical file and variable naming.

B. Input and Output Vulnerabilities

Description: Input is data taken from sources external to the program including user input, databases, files, and remote sources. Key principles include data validation and representation, encryption/decryption (and pseudorandom numbers), and credential authentication. Common issues associated with input and output vulnerabilities include SQL injections, cross-site scripting issues, and backdoor access.

C. Non-input Run-time Vulnerabilities

Description: Non-input run-time vulnerabilities are vulnerabilities that are present during the normal program execution. Key principles include dynamic memory management, error/exception handling, and code obfuscation. Common issues associated with non-input run-time vulnerabilities include logic errors and buffer overflow, memory corruption, and privilege violations.

D. Communication Vulnerabilities

Description: Communication is the interaction of the program with other programs, devices, networks and humans. Key principles include concurrency, secure inter-process communication and authorization, authentication and verification. Common issues include sockets, RPCs, client and server side programming, and communication and networking protocols, such as COM, DCOM and CORBA.

E. Unit Security Testing

Description: Testing is a set of techniques for identifying security flaws in code. Key principles include security test plans, designing test vectors, threat modeling and their integration into the software design process. Common issues associated with testing include the size of the test space and the parochialism of the test process, i.e., it is extremely difficult to prove that a piece of code is 100% secure.

F. Code Reuse

Description: Code reuse is the reuse of mature, verified code in new programs and the design of programs with future use by others in mind. Key principles include decoupling functional and nonfunctional properties and model based techniques for separation of concerns. Common issues associated with code reuse include inherent uncovered security flaws due to change of context and local versus global security.

3.3.2 Complete List of Student Learning Outcomes

The working group identified twenty-five student learning outcomes associated with secure coding.

1. Identify and define common terms relating to secure coding
2. State and discuss principles and techniques relating to secure coding
3. Differentiate between secure coding and patching and explain the advantage of using secure coding techniques
4. Identify common security defects in software and describe their potential impact
5. Explain the role secure coding plays in the secure software development process
6. Analyze the relationship between secure coding and confidentiality, integrity and availability (CIA)
7. Identify and explain common programming techniques for data protection
8. Design and code programs that offer a high level of data protection
9. **Design and code** programs that use file access controls
10. **Explain** authorization, authentication and verification as they relate to secure coding
11. **Identify** and **explain** techniques for preventing input and output vulnerabilities
12. **Design and code** programs that use encryption and decryption algorithms
13. **Analyze** and **design** the parameters in known encryption and decryption algorithms
14. **Design and code** programs that check/validate input and output
15. **Design and code** programs that include dynamic memory management
16. **Design and code** programs that include error and exception handling
17. **Design and code** programs that use inter-process communication
18. **Design and code** programs that use inter-device networking and communication protocols
19. **Describe** race conditions and their impact on programs
20. **Design and code** programs that use concurrency securely
21. **Describe** the results of errors identified by software security tools
22. **Analyze** security risks of various software applications
23. **Construct** a security test plan and provide a **critique** of it
24. **Discuss** the security risks associated with reusable components
25. **Evaluate** the tradeoffs between the local nature of reusable components and the global nature of security

The working group identified seven of the student learning outcomes as core, i.e., being required for all IA programs irrespective of where they are situated.

a. **Identify** and **define** common terms relating to secure coding
b. **State** and **discuss** principles and techniques relating to secure coding
c. **Differentiate** between secure coding and patching and **explain** the advantage of using secure coding techniques
d. **Identify** common security defects in software and **describe** their potential impact
e. **Explain** the role secure coding plays in the secure software development process
f. **Analyze** the relationship between secure coding and confidentiality, integrity and availability (CIA)
g. **Explain** authorization, authentication and verification as they relate to secure coding

The remaining eighteen student learning outcomes are considered to be elective.

h. **Identify** and **explain** common programming techniques for data protection

Table 2 summarizes the complete list of learning outcomes and for each outcome provides a framework for a suggested assessment rubric.
Table 2: Learning outcomes for secure coding topics with assessment rubric frameworks.

<table>
<thead>
<tr>
<th>Core or Elective</th>
<th>Mapping to Secure Coding Topics [Bloom’s Taxonomy Level]</th>
<th>Student Learning Outcome</th>
<th>Poor</th>
<th>Good</th>
<th>Excellent</th>
<th>Assessed by</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Core</td>
<td>A, B, C, D, E, F [knowledge]</td>
<td>Identify and define common terms relating to secure coding</td>
<td>Correctly identifies and defines only a few or no common terms</td>
<td>Correctly identifies and defines most common terms</td>
<td>Correctly identifies and defines all or almost all common terms</td>
<td>Matching, Multiple choice, short answer</td>
</tr>
<tr>
<td>2 Core</td>
<td>A, B, C, D, E, F [knowledge; comprehension]</td>
<td>State and discuss principles and techniques relating to secure coding</td>
<td>Correctly states and discusses only a few or no principles</td>
<td>Correctly states and discusses most principles</td>
<td>Correctly states and discusses all or almost all principles</td>
<td>Matching, short answer, group discussion, oral presentation, student writing</td>
</tr>
<tr>
<td>3 Core</td>
<td>A, B, C, D, E, F [comprehension]</td>
<td>Differentiate between secure coding and patching and explain the advantage of using secure coding</td>
<td>Does not differentiate between secure coding and patching and poorly explains the advantages</td>
<td>Provides some of the differences between secure coding and patching and explains some of the advantages</td>
<td>Thoroughly differentiates between secure coding and patching and provides detailed explanations of the advantages of using secure coding</td>
<td>Student writing</td>
</tr>
<tr>
<td>4 Core</td>
<td>A, B, C [knowledge; comprehension]</td>
<td>Identify common security defects in software and describe their potential impact</td>
<td>Correctly identifies and describes only a few or no security defects</td>
<td>Correctly identifies and describes most security defects</td>
<td>Correctly identifies and describes all or almost all security defects</td>
<td>Matching, short answer, group discussion, oral presentation, writing activity</td>
</tr>
<tr>
<td>5 Core</td>
<td>C, E, F [comprehension]</td>
<td>Explain the role secure coding plays in the secure software development process</td>
<td>Does not or poorly explains the role secure coding plays in the secure software development process</td>
<td>Adequately explains the role secure coding plays in the secure software development process</td>
<td>Thoroughly explains the role secure coding plays in the secure software development process</td>
<td>Writing activity</td>
</tr>
<tr>
<td>6 Core</td>
<td>A, B [analysis]</td>
<td>Analyze the relationship between secure coding and confidentiality, integrity and availability</td>
<td>Only displays understanding of the relationships between secure coding and CIA at the comprehension level</td>
<td>Adequately analyzes the relationships between secure coding and one or two of CIA</td>
<td>Thoroughly analyzes all the relationships between secure coding and CIA</td>
<td>Writing activity</td>
</tr>
<tr>
<td>7 Elective</td>
<td>A [knowledge; comprehension]</td>
<td>Identify and explain common programming techniques for data protection</td>
<td>Correctly identifies and explains only a few or no programming techniques for data protection</td>
<td>Correctly identifies and explains most programming techniques for data protection</td>
<td>Correctly identifies and explains all or almost all programming techniques for data protection</td>
<td>Matching, short answer, group discussion, oral presentation, writing activity</td>
</tr>
<tr>
<td>8 Elective</td>
<td>A [synthesis]</td>
<td>Design and code programs that offer a high level of data protection</td>
<td>Does not code a program that offer a high level of data protection</td>
<td>Successfully codes a program that offer a high level of data protection</td>
<td>Successfully designs and codes a program that offer a high level of data protection</td>
<td>Programming activity</td>
</tr>
<tr>
<td>9 Elective</td>
<td>A [application; synthesis]</td>
<td>Design and code programs that use file access controls</td>
<td>Does not code a program that uses file access controls</td>
<td>Successfully codes a program that uses file access controls</td>
<td>Successfully designs and codes a program that uses file access controls</td>
<td>Programming activity</td>
</tr>
<tr>
<td>10 Core</td>
<td>B, D [comprehension]</td>
<td>Explain authorization,</td>
<td>Does not or poorly explains</td>
<td>Adequately explains one or</td>
<td>Thoroughly explains all of</td>
<td>Writing activity</td>
</tr>
</tbody>
</table>
| Elective | B
<p>| Knowledge; comprehension |
| --- | --- |
| <strong>11</strong> Elective | <strong>Identify and explain</strong> techniques for preventing input and output vulnerabilities | <strong>Correctly identifies and explains</strong> only a few or no techniques for preventing input and output vulnerabilities | <strong>Correctly identifies and explains</strong> most programming techniques for preventing input and output vulnerabilities | <strong>Correctly identifies and explains</strong> all or almost all programming techniques for preventing input and output vulnerabilities | <strong>Matching, short answer, group discussion, oral presentation, Writing activity</strong> |
| <strong>12</strong> Elective | <strong>Design and code</strong> programs that use encryption and decryption algorithms | <strong>Does not code</strong> a program that uses encryption and decryption | <strong>Successfully codes</strong> a program that uses encryption and decryption | <strong>Successfully designs and codes</strong> a program that uses encryption and decryption | <strong>Programming activity</strong> |
| <strong>13</strong> Elective | <strong>Analyze and design</strong> the parameters in known encryption and decryption algorithms | <strong>Does not successfully design</strong> parameters in known encryption and decryption algorithms | <strong>Successfully analyzes or designs</strong> parameters in known encryption and decryption algorithms | <strong>Successfully analyzes and designs</strong> parameters in known encryption and decryption algorithms | <strong>Programming activity; quantitative design problems</strong> |
| <strong>14</strong> Elective | <strong>Design and code</strong> programs that check/validate input and output | <strong>Does not code</strong> a program that checks/validates input and output | <strong>Successfully codes</strong> a program that checks/validates input and output | <strong>Successfully designs and codes</strong> a program that checks/validates input and output | <strong>Programming activity</strong> |
| <strong>15</strong> Elective | <strong>Design and code</strong> programs that include dynamic memory management | <strong>Does not code</strong> a program that includes dynamic memory management successfully | <strong>Successfully codes</strong> a program that includes dynamic memory management | <strong>Successfully designs and codes</strong> a program that includes dynamic memory management | <strong>Programming activity</strong> |
| <strong>16</strong> Elective | <strong>Design and code</strong> programs that include error and exception handling | <strong>Does not code</strong> a program that includes error and exception handling successfully | <strong>Successfully codes</strong> a program that includes error and exception handling | <strong>Successfully designs and codes</strong> a program that includes error and exception handling | <strong>Programming activity</strong> |
| <strong>17</strong> Elective | <strong>Design and code</strong> programs that use inter-process communication | <strong>Does not code</strong> a program that uses inter-process communication successfully | <strong>Successfully codes</strong> a program that uses inter-process communication | <strong>Successfully designs and codes</strong> a program that uses inter-process communication | <strong>Programming activity</strong> |
| <strong>18</strong> Elective | <strong>Design and code</strong> programs that use inter-device networking and communication protocols | <strong>Does not code</strong> a program that uses inter-device networking and communication protocols successfully | <strong>Successfully codes</strong> a program that uses inter-device networking and communication protocols | <strong>Successfully designs and codes</strong> a program that uses inter-device networking and communication protocols | <strong>Programming activity</strong> |
| <strong>19</strong> Elective | <strong>Describe</strong> race conditions and their impact on programs | <strong>Does not describe</strong> race conditions successfully | <strong>Successfully describes</strong> race conditions | <strong>Successfully describes</strong> race conditions and their impact on programs | <strong>Writing activity</strong> |
| <strong>20</strong> Elective | <strong>Design and code</strong> programs that use concurrency securely | <strong>Does not code</strong> a program that uses concurrency securely | <strong>Successfully codes</strong> a program that uses concurrency securely | <strong>Successfully designs and codes</strong> a program that uses concurrency securely | <strong>Programming activity</strong> |</p>
<table>
<thead>
<tr>
<th>Elective</th>
<th>Code</th>
<th>[comprehension]</th>
<th>[analyze]</th>
<th>[synthesis; evaluation]</th>
<th>[comprehension]</th>
<th>[evaluation]</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>E</td>
<td>Describe the results of errors identified by software security tools</td>
<td>Does not describe the results of errors identified by a software security tool</td>
<td>Describes the results of errors identified by a software security tool</td>
<td>Describes and corrects the results of errors identified by a software security tool</td>
<td>Programming activity</td>
</tr>
<tr>
<td>22</td>
<td>E</td>
<td>Compares and contrasts security risks of various software applications</td>
<td>Identifies the security risks in various software applications</td>
<td>Discusses and explains the security risks in various software applications</td>
<td>Compares and contrasts the security risks in various software applications</td>
<td>Writing activity; oral presentation</td>
</tr>
<tr>
<td>23</td>
<td>E</td>
<td>Construct security test plans and provide a critique of it</td>
<td>Does not construct a security test plan successfully</td>
<td>Successfully constructs a security test plan successfully</td>
<td>Successfully constructs and critiques a security test plan</td>
<td>Group activity; written activity</td>
</tr>
<tr>
<td>24</td>
<td>F</td>
<td>Discuss the security risks associated with reusable components</td>
<td>Does not adequately discuss the security risks associated with reusable components</td>
<td>Adequately discusses the security risks associated with reusable components</td>
<td>Thoroughly discusses the security risks associated with reusable components</td>
<td>Writing activity; oral presentation</td>
</tr>
<tr>
<td>25</td>
<td>F</td>
<td>Evaluate the tradeoffs between the local nature of reusable components and the global nature of security</td>
<td>Identifies tradeoffs between reusable components and security</td>
<td>Illustrates tradeoffs between reusable components and security</td>
<td>Successfully evaluates the tradeoffs between the local nature of reusable components and the global nature of security</td>
<td>Writing activity; oral presentation</td>
</tr>
</tbody>
</table>
3.3.3 Discussion

The working group is still evaluating the decision to make student learning outcomes required or elective versus making topics required or elective. On one hand, choosing outcomes to be required or elective necessitates that a detailed subject description be created for all subjects in the areas prior to having a total number of core hours. On the other hand, doing so would seem to provide a more precise matching of core hours to content coverage.

3.3.4 Bibliography

In preparing this secure coding description, the working group was informed by several books, articles, web sites, course syllabi, and curricular guidelines. They include the following:

Books and articles

Websites
2) http://cwe.mitre.org – Common Weakness Enumeration
3) http://cwe.mitre.org – Common Vulnerabilities and Exposures
4) http://sec.mitre.org – Common Configuration Enumeration
5) http://nvd.nist.gov/ - National Vulnerabilities Database
6) http://nvd.nist.gov/cvss-cfm - NVD Common Vulnerabilities Scoring System
7) http://www.securecoding.cert.org/ - CERT Secure Coding Standards
8) http://www.cert.org/secure-coding/ - CERT Secure Coding

Secure coding syllabi
2) http://www.cs.purdue.edu/homes/cs390s/index.html
3) http://www.cse.msstate.edu/~allen/cse8990sp07/syllabus.htm
4) https://www.securecoding.cert.org/confluence/display/s c/15392-Secure+Programming
5) http://faculty.washington.edu/markk/node/197
8) http://www.cse.scu.edu/~tschwarz/COEN296A_06/syllabus.html

Curricular guidelines
1) Information Technology 2008, Curriculum Guidelines for Undergraduate Degree Programs in Information Technology [7]

4. CHALLENGES AND FUTURE DIRECTIONS

It is clear from the previous sections that IA has matured considerably over the past three decades and is now being recognized as its own discipline. The number of faculty and programs that identify themselves as participating in IA education and research continues to grow, and there is an increasing body of scholarly work in IA. This working group has started the task of creating a set of curricular guidelines towards ultimately creating a document similar to CC2001 [4], IT2008 [7] and the other computing curricular guidelines. However, there are many challenges.

First is the matter of completing the set of curricular guidelines. This working group has proposed 11 areas that constitute IA education, and their associated 83 subjects. Additionally, it has provided a detailed description of the secure coding subject. While this description of secure coding can serve as a model for descriptions of the other 82 subjects, significant work is required to actually create and complete the descriptions of those 82 subjects.

In addition to completing the set of curricular guidelines, it would be useful to have specific guidelines for IA programs located...
departments. The majority of IA programs are housed within computing departments and have in the broader IA endeavor.

5. CONCLUSION

Information Assurance is a serious worldwide concern of governments, industry, and academia. The purpose of this working group, and the document produced has been to start the process for creating a set of guidelines for IA programs and for what students need to know upon graduating from IA programs. In defining the areas (and the subjects that make up the areas) that constitute IA, this paper has taken a broad view of IA education encompassing two-year, four-year and graduate degree programs across a variety of academic departments. The paper presents a description of one subject, and the topics and student learning outcomes that make up that subject. In doing so, the paper proposes a model for all of the other subjects, one built from CC2001 [4] but expanded by the addition of an assessment rubric and a detailed description of the topics.

In conclusion, this paper has laid the groundwork in support of what will eventually become IA2013, a set of guidelines for IA education. Much more work remains to be accomplished. Additionally, the authors of this working group recognize that while the future of IA education is likely quite bright, there are many challenges facing the emerging discipline of Information Assurance education.

6. ACKNOWLEDGMENT

This material is partially based upon work supported by the National Science Foundation under Grant No. 1023963. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

7. REFERENCES


### Appendix A. Areas and Subjects of IA Curriculum

<table>
<thead>
<tr>
<th>Areas</th>
<th>Subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fundamental Concepts</strong></td>
<td></td>
</tr>
<tr>
<td>History</td>
<td>Basic Terminology</td>
</tr>
<tr>
<td></td>
<td>Information Assurance Concepts</td>
</tr>
<tr>
<td></td>
<td>Awareness</td>
</tr>
<tr>
<td></td>
<td>Industry/Government Guidelines</td>
</tr>
<tr>
<td></td>
<td>National/Cultural Differences</td>
</tr>
<tr>
<td><strong>Cryptography</strong></td>
<td>Applications of Cryptography</td>
</tr>
<tr>
<td></td>
<td>Mathematical Preliminaries</td>
</tr>
<tr>
<td></td>
<td>Cryptography Legal Issues</td>
</tr>
<tr>
<td></td>
<td>Basic Cryptography Terminology</td>
</tr>
<tr>
<td></td>
<td>Formal Definitions</td>
</tr>
<tr>
<td></td>
<td>Historical Methods</td>
</tr>
<tr>
<td></td>
<td>Information Theory</td>
</tr>
<tr>
<td></td>
<td>Cryptographic Primitives</td>
</tr>
<tr>
<td></td>
<td>Cryptanalysis</td>
</tr>
<tr>
<td></td>
<td>Cryptographic Algorithm Design</td>
</tr>
<tr>
<td></td>
<td>Common Protocols</td>
</tr>
<tr>
<td></td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td><strong>Security Ethics</strong></td>
<td></td>
</tr>
<tr>
<td>Privacy Issues</td>
<td></td>
</tr>
<tr>
<td>Hacking and Cracking</td>
<td></td>
</tr>
<tr>
<td>Legal Issues</td>
<td></td>
</tr>
<tr>
<td>Ethical Dilemmas</td>
<td></td>
</tr>
<tr>
<td>National/Cultural Differences</td>
<td></td>
</tr>
<tr>
<td>Challenges</td>
<td></td>
</tr>
<tr>
<td>Codes of Ethics/Professionalism</td>
<td></td>
</tr>
<tr>
<td><strong>Security Policy</strong></td>
<td></td>
</tr>
<tr>
<td>Strategies and Plans</td>
<td></td>
</tr>
<tr>
<td>Policies/Guidelines/Standards/Best Practices</td>
<td></td>
</tr>
<tr>
<td>Procedures</td>
<td></td>
</tr>
<tr>
<td>Privacy Policies</td>
<td></td>
</tr>
<tr>
<td>Compliance</td>
<td></td>
</tr>
<tr>
<td>Enforcement</td>
<td></td>
</tr>
<tr>
<td>Formal Policy Models</td>
<td></td>
</tr>
<tr>
<td>Policy Languages</td>
<td></td>
</tr>
<tr>
<td>Notions of Security</td>
<td></td>
</tr>
<tr>
<td><strong>Digital Forensics</strong></td>
<td></td>
</tr>
<tr>
<td>Basic Principles and Methodology</td>
<td></td>
</tr>
<tr>
<td>Computer/Network/System Attacks</td>
<td></td>
</tr>
<tr>
<td>Rules of Evidence</td>
<td></td>
</tr>
<tr>
<td>Search and Seizure</td>
<td></td>
</tr>
<tr>
<td>Digital Evidence</td>
<td></td>
</tr>
<tr>
<td>Data Analysis and Validation</td>
<td></td>
</tr>
<tr>
<td>Preservation of Data</td>
<td></td>
</tr>
<tr>
<td>Legal and Reporting Issues</td>
<td></td>
</tr>
<tr>
<td><strong>Areas</strong></td>
<td>Subjects</td>
</tr>
<tr>
<td><strong>Access Control</strong></td>
<td>Basic Principles</td>
</tr>
<tr>
<td></td>
<td>Physical Access Control</td>
</tr>
<tr>
<td></td>
<td>Technical / System Access Control</td>
</tr>
<tr>
<td><strong>Security Architecture and Systems</strong></td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td>Implementation</td>
</tr>
<tr>
<td></td>
<td>Usability</td>
</tr>
<tr>
<td></td>
<td>System Threats and Vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>Operating System Security</td>
</tr>
<tr>
<td></td>
<td>Multi-level/multi-lateral</td>
</tr>
<tr>
<td></td>
<td>Design and Testing</td>
</tr>
<tr>
<td></td>
<td>Penetration Testing</td>
</tr>
<tr>
<td></td>
<td>Products</td>
</tr>
<tr>
<td><strong>Network Security</strong></td>
<td></td>
</tr>
<tr>
<td>Network Basics</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Protocols</td>
</tr>
<tr>
<td></td>
<td>Standards</td>
</tr>
<tr>
<td></td>
<td>Vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>Defense Mechanisms/Countermeasures</td>
</tr>
<tr>
<td></td>
<td>Network Design</td>
</tr>
<tr>
<td></td>
<td>Network Auditing</td>
</tr>
<tr>
<td></td>
<td>Penetration Testing</td>
</tr>
<tr>
<td><strong>Risk Management</strong></td>
<td></td>
</tr>
<tr>
<td>Risk Analysis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cost/Benefit Analysis</td>
</tr>
<tr>
<td></td>
<td>Continuity Planning</td>
</tr>
<tr>
<td></td>
<td>Defense In Depth</td>
</tr>
<tr>
<td></td>
<td>Contingency and Disaster Recovery</td>
</tr>
<tr>
<td></td>
<td>Security Audit</td>
</tr>
<tr>
<td></td>
<td>Asset Management</td>
</tr>
<tr>
<td></td>
<td>Risk Communication</td>
</tr>
<tr>
<td></td>
<td>Enforcement</td>
</tr>
<tr>
<td><strong>Attacks / Defenses</strong></td>
<td></td>
</tr>
<tr>
<td>Threats and Vulnerabilities</td>
<td></td>
</tr>
<tr>
<td>Types of Attacks</td>
<td></td>
</tr>
<tr>
<td>Types of Attackers</td>
<td></td>
</tr>
<tr>
<td>Defense Mechanisms</td>
<td></td>
</tr>
<tr>
<td>Incident Response</td>
<td></td>
</tr>
<tr>
<td><strong>Secure Software Design and Engineering</strong></td>
<td></td>
</tr>
<tr>
<td>Secure Software Specification</td>
<td></td>
</tr>
<tr>
<td>Secure Coding</td>
<td></td>
</tr>
<tr>
<td>Secure Testing</td>
<td></td>
</tr>
<tr>
<td>Program Verification and Simulation</td>
<td></td>
</tr>
<tr>
<td>Language-based Security</td>
<td></td>
</tr>
<tr>
<td>Secure Design</td>
<td></td>
</tr>
<tr>
<td>Maintenance</td>
<td></td>
</tr>
</tbody>
</table>