ESTABLISHING AN INFORMATION ASSURANCE CURRICULA BASELINE WITHIN EXISTING COMPUTER SCIENCE, SOFTWARE ENGINEERING, AND INFORMATION TECHNOLOGY PROGRAMS

by

Aaron E Wampach

STEVEN BROWN, PhD, Faculty Mentor and Chair

RICHARD LIVINGOOD, PhD, Committee Member

GARY STROEBEL, PhD, Committee Member

Sue Talley, EdD, Dean, School of Business and Technology

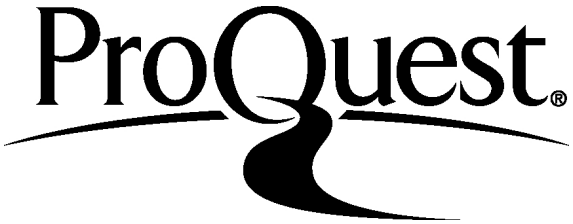
A Dissertation Presented in Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

Capella University

July 2015



Abstract

Many colleges and universities have started to offer information assurance (IA) degrees. Existing computer science (CS), software engineering (SE), and information technology (IT) programs need to also be updated to address current and future information assurance needs. There is still a need for specialized computer scientists, software engineers, and information technologists, but students also need to understand the various IA needs required by all organizations today. The problem is how colleges and universities should address covering IA concepts in existing CS, SE, and IT curriculums. This research addresses what IA concepts should be taught as core concepts and what IA concepts should be integrated or repeated through several classes to help retain the IA concept. Perhaps more importantly, this research also addresses which existing IA concepts should be removed from CS, SE, and IT curriculums. This qualitative research utilized a literature review to establish three propositions which were tested. First proposition validated which IA concepts should be incorporated into existing CS programs and how they should be integrated. The second proposition validated which IA concepts should be incorporated into existing SE programs and how they should be integrated. The final proposition validated which IA concepts should be incorporated into existing IT programs and how they should be integrated. Each proposition was validated against ten examined cases. Within the CS proposition, all concepts except Forensics and Risk Management are supported. Within the SE proposition, all concepts are supported. Finally within the IT proposition, all concepts except Forensics are supported. Findings from this research will allow university administrators to better incorporate IA concepts into existing CS, SE, and IT curriculums.

Dedication

This research is dedicated to my lovely wife Mari, and my little bundle of joy Wren. Without the support and patience from both of you, this research would have not been possible.

Acknowledgments

I would like to acknowledge Dr. Brown my mentor for helping through this long process. I would like to thank the rest of my dissertation committee, Dr. Livingood and Dr. Stroebel, without whose guidance this dissertation would have not been completed.

Table of Contents

List of Tables

CHAPTER 1. INTRODUCTION

Introduction to the Problem

Numerous researchers have discussed the need for integrating information assurance topics within existing computer science (CS), software engineering (SE), and information technology (IT), but little research has been completed on what information assurance (IA) topics specifically need to be integrated (Lester, Narang, & Chen, 2008; Lester, 2010; Bishop & Frincke, 2008; Peltsverger & Karam, 2010). Frank and Werner (2010) state, "Many security experts have consistently championed the philosophy that better design, implementation, and operation of software can prevent many of the security problems that plague ubiquitous computing" (p. 49). A better design needs to be applied to CS, SE, and IT that enhance knowledge of IA topics that are in these curricula.

Today, either a specific set of classes is added to existing CS, SE, or IT curricula to cover topics within IA or an entirely new program is created (Peltsverger & Karam, 2010; Lester, 2010; Lester, Narang, & Chen, 2008). Adding an IA class at the end of a program is not sufficient (Dark, Ekstrom, & Lunt, 2005; Lester, Narang, & Chen, 2008). Numerous researchers support this stance of integration of IA within existing curriculum (Schweitzer, Humphries, & Baird, 2006; Lester, 2010; Taylor & Azadegan, 2006; Lester, Narang, & Chen, 2008). IA topics need to be a requirement in all CS, SE, and IT curricula (Karam & Peltsverger, 2009; Reynolds & Goda, 2007). Colleges and universities today treat IA as a separate topic within existing CS, SE, and IT curricula (Taylor & Azadegan, 2006; Hjelmås & Wolthusen, 2006). This lack of understanding basic IA topics carries over to real world systems and drastically reduces the confidentiality, integrity, and availability of all systems; costing organizations $180

billion (US) in system outages and data breaches in 2007 alone (Rice, 2007). Researchers have suggested that teaching IA topics in CS, SE and IT can mitigate the risks created everyday by a lack of knowledge within computer systems, applications and processes (Rogers, 2006; Kabay, 2005).

Background of the Study

Information system education has evolved since its inception (Dark, 2004; Gupta, 2007). The Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE) Computer Society were formed in the mid-1940s (ACM, 2009; IEEE, 2009) from theories of computation. The roots of all of these programs come from the study of mathematics and computational automation (Gupta, 2007). IA was never considered. Early studies stemmed from automating computation and not from the processing of data (Cooper, 2005). From the field of CS a specialty emerged in the study of software development. SE started to examine the efficiency and effectiveness of computer algorithms, expanding the field further from strictly computation to more data processing. Gradually the need expanded further to integrate multiple systems and incorporate business processes, which lead to the creation of the field of IT. This gradual progression from computation to information never incorporated concepts of IA. Today IA is a requirement for many different organizations, especially organizations governed by one or more of the many regulations that exist globally (Ferrara, 2006).

As educational programs evolved different classes were added to incorporate these new concepts into existing curriculum. For example, initial programs were developed around computational theory and development languages. Structured

development evolved along with the creation of different data structures (Gupta, 2007). Course work was added to study these new techniques and a new program of SE was developed. As distributed systems were developed additional classes on networking theory also emerged. More specialty classes began to emerge and another program of IT was established (Gupta, 2007).

Today, another set of classes is being established, which is defining a new program of study known as information assurance (IA). IA as an independent program has evolved in different ways depending on the university and college. Some universities and colleges have added new classes, while others have created entirely new programs (Schweitzer, Humphries, & Baird, 2006; Taylor & Azadegan, 2006). This research examined what IA topics should be integrated into existing CS, SE, and IT programs and curricula.

Several studies have been compiled evaluating the integration of IA concepts within CS, SE, and IT programs (Conti, Hill, Lathrop, Alford, & Ragsdale, 2003; Lester, 2010; Bellovin et al., 2008; Bishop & Frincke, 2008; Peltsverger & Karam, 2010; Bratus, Shubina, & Locasto, 2010; Uzubell, Liles, & Jiang, 2010). Often classes teaching IA curricula are appended to the end of a program when all the other class work was completed (Lester, 2008; Lester, Narang, & Chen, 2008; Bishop & Frincke, 2008). Weaknesses in software development, system integrations and computational processing are some of the core issues with all data breaches and systems outages experienced today (OWASP, 2007).

Statement of the Problem

Organizations are demanding students with skills in information assurance (Ponemon Institute, 2014). Computer science, software engineering, or information technology programs lack a sufficient integration of IA concepts to communicate to students. Numerous researchers (Peltsverger & Karam, 2010; Bishop & Frincke, 2008; Lester, 2010; Lester, Narang, & Chen, 2010) support the claim that colleges and universities need to integrate IA curricula into CS, SE and IT programs. Unlike in the past where specialty programs evolved from existing programs, IA has become a fundamental concept, which needs to be incorporated into all CS, SE and IT programs (Schweitzer, Gibson, & Collins, 2009; Lester, Narang, & Chen, 2008; Lester, 2010; Taylor & Azadegan, 2006).

Numerous researchers have discussed the need to integrate IA within CS, SE, and IT curricula, while other researchers have argued a dedicated program was required (Dark, Ekstrom, & Lunt, 2005; Geoghegan, 2008; Goel et al., 2006). Little research has been compiled that evaluates what topics specifically within IA should be addressed in CS, SE, or IT programs and if these IA topics align with business requirements (Lester, Narang, & Chen, 2008; Neubauer, Klemen, & Biffl, 2006; Xiaobin, Yong, & Hongsheng, 2007). The problem this study has examined was the lack of IA topics being taught within existing CS, SE, and IT curricula (Myers & Riela, 2008).

Purpose of the Study

The purpose of this qualitative case study was to determine what IA topics should be taught within CS, SE, and IT curricula. Findings from this research may assist university administrators to better incorporate IA curricula into existing CS, SE, and IT programs and curriculum.

Rationale

There are several reasons why a literature review was evaluated against several qualitative case studies. The literature review shows various studies on the integration of IA concepts within CS, SE, and IT programs, but there was very little research on the specific IA competencies that should be included specifically within CS, SE, and IT. Secondly, the literature review showed a bias towards quantitative research by quantifying the need for IA integration within CS, SE, and IT. Little research had been compiled utilizing the qualitative perspective of capturing experts' opinions on IA integration within CS, SE, and IT curriculum.

Different IA competencies need to be addressed in different curricula. Organizations are hiring recent computer science, software engineering, and information technology graduates, but are the expectations of the hiring managers being met. This research utilized a collection of qualitative case studies to capture the perceptions of these hiring managers on information assurance competencies and compared them against the competencies addressed in the literature review.

Research Question

This study answered the following research question:

Research question

What information assurance core competencies should be included in computer science, software engineering, and information technology curricula?

Significance of the Study

This research was significant because there was a deficiency of research available on what IA concepts should be integrated within CS, SE, or IT. Numerous studies (Petrova, Kaskenpalo, Philpott, & Buchan, 2004; Bogolea & Wijekumar, 2004; Taylor & Azadegan, 2006; Wang, 2008) have been compiled regarding the need to integrate IA curricula into existing programs, but little research had been compiled on the what IA concepts specifically needed to be incorporated into existing CS, SE, and IT coursework and programs. This research evaluated what IA concepts should be integrated into CS, SE, and IT and what concepts should be taught within an independent IA program (Petrova, Kaskenpalo, Philpott, & Buchan, 2004; Bogolea & Wijekumar, 2004; Taylor & Azadegan, 2006; Wang, 2008; Meiselwitz, 2008; Goel et al., 2006).

The review of literature suggested the need to integrate IA curriculum within CS, SE, and IT. What the literature did not distinctly show was what IA concepts should be taught purely within IA and which concepts should be presented within CS, SE, or IT. Most of the current literature that focuses on IA concepts (ACM & IEEE, 2008a; Tipton, 2009) is several years old now and do not address current concerns, such as cloud computing.

Definition of Terms

Information Assurance. Information assurance is the assertion for the availability, integrity, authentication, confidentiality, and non- repudiation of information systems. This assurance also includes the restoration of information systems by incorporating protection, detection, and reaction mechanisms (CNSS, 2006, p.35).

Computer Science. Computer science is the study of the hardware, software, and academic aspects of computing and computing devices. Computer science also is the study of the application of these computing devices to scientific, technological, and business processes and issues (University of Minnesota, 2005).

Software Engineering. Software engineering is the study and the application of engineering, scientific, and mathematical principles and methods to construction of quality software (Humphrey, 1989).

Information Technology. Information technology is the study and application of computer-based information systems, particularly software applications and computer hardware (ITAA, 2008).

Nature of the Study

Research Design

A qualitative case study research design was utilized to address the stated research questions (Yin, 2009). The first section of this research reviewed current research on IA education and reviewed current recommendations from professional organizations. The purpose of this literature review was to determine the lack of information on what should be taught in IA and what IA concepts should be taught within CS, SE, and IT. The second section synthesized a set of qualitative cases consisting of several interviews from information assurance leaders in various business sizes and sectors.

Sampling

Information assurance specialists were randomly selected from organizations ranging from small and medium sized businesses through large Fortune 500 organizations. Numerous business verticals were selected, from healthcare to travel and leisure. This broad sampling helped validate the requirement of IA skills across any size organization or business vertical.

The National Centers of Academic Excellence in Information Assurance Education (CAE/IAE) supported the use of practitioners within the sample. Within the CAE/IAE (2012) criteria, each center was required to incorporate an outreach and collaboration program that provides students with knowledge experiences from IA practitioners. Input from IA practitioners was an important aspect supported by several researchers (Tenenberg, 2009; Dahlberg, Barnes, Buch, & Rorrer, 2011). Input from IA practitioners will help colleges and universities develop programs that are relevant and fill a current knowledge gap in the industry.

Assumptions and Limitations

Assumptions

Two assumptions were inherent in the examination of IA curricula within existing CS, SE, and IT programs. The first assumption was there is a relationship between academic institutions and private organizations. It is incumbent upon academic institutions to teach the skills required by the industry; otherwise they become a purely academic exercise.

A second assumption was that Information Security is an aspect of IA. Many different colleges and universities may either have an Information Security program or an IA program. For the sake of this research, Information Security and IA will be considered in the same context.

Limitations

Interviewing every possible size and business vertical would have been impractical, time consuming and expensive. This research limited itself to only interview a total of ten information assurance leaders. This limitation was somewhat justified by the pure nature of a qualitative research method (Yin, 2009). According to Yin (2009), case studies rely on analytic generalization, whereas surveys rely on statistical generalization. Another possible limitation was that current information assurance leaders might not understand what information assurance concepts exist today.

Organization of the Remainder of the Study

The remainder of the research was broken down into four different chapters. Chapter two is a literature review. The literature review established the purpose of this research. Part one of the literature review evaluated the current recommendations of the inclusion of IA curricula within CS, SE, and IT. Part two of the literature review established the need for repetition of IA curricula throughout a program. Part three of the literature review examined the need for collaboration with organizations in determining the specific requirements of IA within CS, SE, and IT. Part four of the literature review examined current research on the evaluation of the current workforce to validate if a gap in IA knowledge exists today. The final section in the literature review evaluated current research on the integration on IA curricula within CS, SE, and IT. Chapter three further describes the research methodology in detail. Chapter four presents the analysis of data and finally, chapter five presents the research findings and conclusions.

CHAPTER 2. LITERATURE REVIEW

Overview

This chapter examines the current state of information assurance within computer science, software engineering, and information technology, as well as the current state of IA education. Supporting evidence and research was evaluated that supports the proposed research question. The first section examines the current state of IA curriculum within CS. The second section explores the current state of IA within SE. The third section examines the current state of IA within IT. The fourth section compares and contrasts the curriculum being taught in IA, and determines what differentiates IA from CS, SE, and IT. The fifth section examines current research on the need to integrate IA within CS, SE, and IT. The final section recommends changes to CS, SE, and IT curricula to successfully integrate IA concepts within each program.

Current State of Computer Science

Two approaches are typically taken when integrating IA curriculum within CS (Morneau, 2004; Irvine, Chin, & Frincke, 1998). The first approach is to fully integrate IA concepts into each and every CS class. The second approach is to append IA classes into an existing CS curriculum. A majority of the research today examines the integration of IA specialization classes within existing CS curricula (Petrova, Kaskenpalo, Philpott, & Buchan, 2005; Crowley, 2003; Irvine & Nguyen, 2010; Chatmon, Chi, & Davis, 2010; Ghafarian, 2007).

Many CS programs today contain several specialty tracks (Kamali, Liles, Winer, Jiang, & Nicolai, 2005). Many colleges and universities choose to offer an information

security track within CS (Perez et al., 2011). This security specialization track is generally broken down into several core classes: information security, cryptography, forensics, network security, ethics, incident handling, and enterprise security architecture (Petrova, Kaskenpalo, Philpott, & Buchan, 2005; Bogolea & Wijekumar, 2004; Abi-Antoun & Barnes, 2010; Blackwell, 2009). These specializations have been defined for several years and have changed very little since their definition.

A common trend in security specialization tracks within CS is to begin with an awareness course, usually an introductory information security track (Markham, 2009; Cooper et al., 2010; Bhagyavati, Naugler, & Frank, 2005). The contents of the curriculum vary depending on the college or university and how many classes are being offered in the security specialization track. Sometimes this class may contain concepts of cryptography, ethics, incident handling, network security, and other key competencies. Theoharidou and Gritzalis (2007) define a common body of knowledge which information security should entail. An information security course should contain the following content: access control and privacy (Cate, 2009); risk and attacks (Clark, Singleton, Tyree, & Hale, 2008; Foley, 2009); cryptography; networks; security design; business; and ethics and law (Theoharidou & Gritzalis, 2007; Losavio, Shutt, & Keeling, 2010). Some interesting addendums included by Theoharidou and Gritzalis (2007) include privacy, risk and business (Foley, 2009). These concepts defined the need for IA and the need to protect both the confidentiality and availability of data. It also showed the evolution away from information security to IA.

Cryptography is the next security class typically offered in many IA specialization tracks within CS (Cooper et al., 2010; Perez et al., 2011; Hamilton, Owor, & Dajani,

2009; Jensen, Cline, & Guynes, 2006; Lester, Narang, & Chen, 2008; Streff & Zhou, 2005; Schweitzer, Humphries, & Baird, 2006; Crowley, 2003; Pothamsetty, 2005; Geoghegan, 2008; Hjelmas & Wolthusen, 2006; Al-Hamdani, 2006; Markham, 2009; Bhagyavati, Naugler, & Frank, 2005). The need to understand cryptography is further supported by several security organizations and frameworks (CNSS, SANS, $(ISC)^2$, ACM and IEEE). Topics that need to be covered in cryptography include: symmetric and asymmetric encryption; digital signatures; cryptographic protocols, such as key exchange; cryptanalysis; and steganography (Pothamsetty, 2005; Lester, Narang, & Chen, 2008; Ferguson, Schneier, & Kohno, 2010). The need for a standalone cryptography class is warranted in CS, where analysis of algorithms and technique are part of the underlying framework, as defined by ABET (2012) program criteria for CS Ferguson, Schneier, & Kohno, 2010). Cryptography on the other hand within IT could probably be integrated within an integrated information security class that covers multiple security concepts.

Forensics is the next common course that was integrated into a CS curriculum as a specialized security track (Dimkov, Pieters, & Hartel, 2011; Cooper, Finley, & Kaskenpalo, 2010; Perez et al., 2011; Jensen, Cline, & Guynes, 2006; Cooper et al., 2010; Dark, Ekstrom, & Lunt, 2005; Lester, Narang, & Chen, 2008; Cooper, 2005; Streff & Zhou, 2005; Geoghegan, 2008; Crowley, 2007; Hjelmas & Wolthusen, 2006; Sexton, 2008; Bhagyavati, Naugler, & Frank, 2005; McGuire & Murff, 2006; Vaughn & Dampier, 2007). This is typically differentiated from physical forensics classes taught in criminology by defining it as digital, computer, or information forensics. Digital forensics is the study of collecting, preserving, and reconstructing stored data evidence. This

includes volatile data, nonvolatile data, and network packet captures from a computer or network where a suspected crime has occurred (Cooper, Finley, & Kaskenpalo, 2010). Depending on the college or university, forensics classes are broken down into two or more separate classes. The inclusion of forensics within a specialty security track in CS does not seem to make sense, depending on the scope of the content. Most if not all of the forensics classes taught today focus on the processes of collecting, preserving and presenting evidence in a forensically sound manner (Cooper, Finley, & Kaskenpalo, 2010), this falls more in line with IT outcomes. On the other hand if the forensics class focused on the algorithms used, such as for file carving, the inclusion would satisfy the learning outcome defined in the ABET (2012) program criteria for CS.

The next IA specialization class defined within CS is network security (Cooper et al., 2010; Perez et al., 2011; Hamilton, Owor, & Dajani, 2009; Jensen, Cline, & Guynes, 2006; Lester, Narang, & Chen, 2008; Streff & Zhou, 2005; Schweitzer, Humphries, & Baird, 2006; Crowley, 2003; Pothamsetty, 2005; Geoghegan, 2008; Hjelmas & Wolthusen, 2006; Al-Hamdani, 2006; Markham, 2009; Bhagyavati, Naugler, & Frank, 2005). Ironically, network security is not fully defined and agreed upon. Network security overlaps with several different concepts, such as cryptography and access control. The ACM and IEEE (2008a) define network security within the CS curriculum as the following: fundamentals of cryptography; authentication protocols; digital signatures; network attack types, such as denial of service, flooding, hijacking, etc; access control mechanisms; basic network defense tools and strategies, such as: intrusion detection, firewalls, Kerberos, IPSEC, virtual private networks (VPN), and network address translation (NAT); network management policies; and auditing and logging

(Carlin & Gallegos, 2007; Pan, 2007). Network security many times was consolidated with systems security, as even defined by CNSS (2008) terms. Yang and Nguyen (2006) have defined a better framework for teaching network security, defining it as the following: asset identification, threat assessment, risk assessment, policy construction, network security design, network security implementation, and audit and improvement. Their framework integrates some of the concepts from the ACM and IEEE (2008) model, but also integrates newer IA concepts, such as: asset identification, threat assessment, risk assessment, and policy construction. These are all critical concepts that need to be addressed and understood by today's college and university CS graduates.

Ethics is the next defined IA specialization class within CS (Cooper et al., 2010; Perez et al., 2011; Hamilton, Owor, & Dajani, 2009; Jensen, Cline, & Guynes, 2006; Lester, Narang, & Chen, 2008; Streff & Zhou, 2005; Schweitzer, Humphries, & Baird, 2006; Crowley, 2003; Pothamsetty, 2005; Geoghegan, 2008; Hjelmas & Wolthusen, 2006; Al-Hamdani, 2006; Markham, 2009; Bhagyavati, Naugler, & Frank, 2005; Losavio, Shutt, & Keeling, 2010). Cooper et al. (2010) define ethics as the following core topics: privacy issues (Cate, 2009); hacking and cracking; legal issues, such as security breaches and misuse; prevalent ethical dilemmas like whistle blowing; national or cultural differences; basis for ethical decision making; challenges in balancing freedom of information and security; security as a societal goal; and legal vs. ethical aspects. The importance of an ethics class that covers all the topics defined by Cooper et al. (2010) cannot be overstated (Schaefer, 2009). Unfortunately many CS programs outside of the US do not require a specialized ethics class (Perez et al., 2011).

Incident handling and response is the next IA specialization class defined within CS (Cooper et al., 2010; Perez et al., 2011; Hamilton, Owor, & Dajani, 2009; Jensen, Cline, & Guynes, 2006; Lester, Narang, & Chen, 2008; Streff & Zhou, 2005; Schweitzer, Humphries, & Baird, 2006; Crowley, 2003; Pothamsetty, 2005; Geoghegan, 2008; Hjelmas & Wolthusen, 2006; Al-Hamdani, 2006; Markham, 2009; Bhagyavati, Naugler, & Frank, 2005). Cooper et al. (2010) defines incident handling in a slightly different perspective, as in attack and defense. This viewpoint captures the intent of IA better by analyzing both preventative and detective controls. Cooper et al. (2010) defines attack and defense as: threats and vulnerabilities, types of attacks, types of attackers, defense mechanisms, and incident response. This somewhat overlaps with network security, but makes better sense to incorporate and analyze both the attack and the response processes at the same time. At the same time there was no global standard for incident response (Tjoa, Jakoubi, Goluch, & Quirchmayr, 2008). The closest defined standard for incident response comes from NIST (2008). Cooper et al. (2010) seem to follow the NIST (2008) SP800-61rev1 definition of incident handling. NIST defines incident handling at a very high level as the following: organizing a computer security incident response capability; handling an incident; handling denial of service incidents; handling malicious code incidents; handling unauthorized access incidents; handling inappropriate usage incidents; and handling multiple components incidents. NIST (2008) covers both the attack and the response. The NIST (2008) incident-handling standard was a nice framework that can easily convert into a full incident management class. Unfortunately the concepts covered within incident handling match better to an IT class than a CS class,

as defined by ABET (2008) program criteria. Incident handling is more about process around people and systems than algorithms and system architecture.

Security architecture is the final IA specialization class within CS (Cooper et al., 2010; Perez et al., 2011; Hamilton, Owor, & Dajani, 2009; Jensen, Cline, & Guynes, 2006; Lester, Narang, & Chen, 2008; Streff & Zhou, 2005; Schweitzer, Humphries, & Baird, 2006; Crowley, 2003; Pothamsetty, 2005; Geoghegan, 2008; Hjelmas & Wolthusen, 2006; Al-Hamdani, 2006; Markham, 2009; Blackwell, 2009). Cooper et al. (2010) defines security architecture as the following: hardware security, implementation issues; usability, identify and analyze system threats and vulnerabilities, operating system security, multi-level and multi-lateral security, design and testing, penetration testing, and product evaluations. Amer and Hamilton (2008) offer a slightly different perspective. They define security architecture as a combination of the following: networks, host components, applications, information, software, hardware, databases, and physical components. One different perspective that Amer and Hamilton (2008) provide was the need for physical security. Whether it was the definitions by Cooper et al. (2010) or Amer and Hamilton (2009), not all aspects directly apply to CS. For example, there is a need to evaluate physical security, but physical security is more in line with IA as an independent program or IT.

Current State of Software Engineering

Similar to CS, integrating IA curriculum into SE can be approached in two ways: integrating IA concepts throughout a SE program, or appending specialty classes into the curriculum (Lester & Jamerson, 2008; Lester & Jamerson, 2009; Mead & Hough, 2006; Conklin & Dietrich, 2007; Rubin & Misra, 2007; Taylor & Azadegan, 2006; Bruschi, De

Win, & Monga, 2006; Payne, 2010). Unlike the research explored in CS, there does not appear to be a clear definition of specific classes that should be appended. The integration of IA curriculum throughout a SE curriculum better aligns with how practitioners align security within the software development lifecycle (Taylor & Azadegan, 2006).

The challenge of integrating IA curriculum within existing SE programs is many SE programs do not have the room to include new concepts (Conklin & Dietrich, 2007). One change that could be made in SE was to incorporate security concepts within the programming examples that are presented throughout an entire class and curriculum. Through the use of repetition, IA concepts can be repeated numerous times within a curriculum, strengthening the understanding and comprehension of security concepts (Conklin & Dietrich, 2007).

According to System Administration, Networking, and Security Institute (SANS) (2011), one of the top security controls is application software security. SANS breaks down application software security into the following: buffer overflows; SQL injection attacks; cross-site scripting; cross-site request forgery; and click jacking of code. The Open Web Application Security Project (OWASP) echoes these same vulnerabilities. OWASP (2010) does include a few different concepts, such as: broken session authentication and session management, insecure direct object references, security misconfigurations, insecure cryptographic storage, failure to restrict URL access, insufficient transport layer protection, and un-validated redirects and forwards. Numerous researchers (Walden, 2008; Park, 2011; Edge & Stamey, 2010) have incorporated the OWASP top 10 into their own research, validating the significance of the OWASP top 10. Between the SANS (2011) top security controls and the OWASP (2010) top 10

vulnerabilities, there are enough security vulnerabilities that can be successfully repeated throughout a SE curriculum. Each development topic and lab activity can emphasize a specific SANS (2011) or OWASP (2010) application vulnerability (Edge & Stamey, 2010).

Unfortunately this is only half of the story. Secure application development is more than addressing vulnerabilities; good development standards need to also be addressed (Taylor & Azadegan, 2006). Taylor and Azadegan (2006) define specific touch points that are fundamental to software development. They are: program defensively; defense in depth; hiding information; least privilege; all input was evil; assume the impossible; deny by default; and design by contact. Taylor and Azadegan (2006) repeat concepts that were defined by the NSTISSI (1994): segregation of duties; concept of least privilege; identification and authentication; access privileges; internal labeling; audit trails and logging; need-to-know controls; and malicious logic protection. The NSTISSI (1994) also defines another concept of configuration management, which includes: programming standards and controls; documentation; and change controls. Very little research has been compiled on programming standards, documentation and change controls within SE (Maqsood & Javed, 2007). These concepts are typically rolled into a software project management course, but even then the concepts that are covered are not completely agreed upon (Maqsood & Javed, 2007).

Rubin and Misra (2007) look at integrating IA into SE by appending classes into the existing curriculum. Rubin and Misra (2007) suggest adding a computer security class as a minimum, with a full security track also including an advanced computer security class, computer forensics, and legal issues in technology class. For the most part, Rubin

and Misra (2007) define the computer security class as a compilation of cryptography concepts. In theory, the class could have been called cryptography. Rubin and Misra (2007) then define advanced computer security as advanced cryptography and analysis of software application vulnerabilities, such as the ones defined by OWASP (2010) and SANS (2011). The last two classes, Computer Forensics and Legal Issues in Technology, are identical to classes with similar titles in CS and information technology. It is unclear why a computer forensics class would be included in a SE curriculum and why the computer security class was not called cryptography.

Lester and Jamerson (2008) take a similar approach to Rubin and Misra (2007) in developing a specific class related to software security. Lester and Jamerson (2008) break down software security into the following objectives: Introduction and the need for software security; security within the software development lifecycle; managing software security risk (Clark, Singleton, Tyree, & Hale, 2008; Foley, 2009; Islam & Dong, 2008); tools and techniques for designing trustworthy software; and special topics within software security. Lester and Jameson (2008) do bring up several good topics, such as software risk management and security within a software development lifecycle. Combing software risk management and security within a software development lifecycle, along with the concepts of configuration management from the NSTISSI (1994) would establish a standalone class of concepts needed by Software Engineers. The combined class could be called software risk and project management.

An ideal integration of IA curriculum into SE would be a hybrid approach, where software security vulnerabilities are integrated throughout the curriculum; while specific

security classes such as: software risk and project management; and cryptography are appended into the existing curriculum.

Current State of Information Technology

Research in IT follows the same trends that are established in CS and SE. IA concepts can either be integrated throughout the IT curriculum or specific IA classes appended into the existing curriculum (Rowe, Lunt, & Ekstrom, 2011; Meiselwitz, 2008; Chow, Chmura, & Linberg, 2007; Reynolds & Goda, 2007). In comparison with CS and SE, IT and IA are more tightly coupled. The best approach of integrating IA curriculum within IT was to use a hybrid approach, by both integrating IA concepts throughout the IT curriculum, and by also appending specific IA classes within the IT Curriculum. This hybrid approach was supported by Rowe, Lunt, and Ekstrom (2011), who state:

> Many academics have stated the need for security-across-the-curriculum in IT Programs. The proposal of cyber-security emphasis should not be seen as countering this research and we caution strongly against removing content from IT topics in order to move it to defined cyber-security courses. The benefits of security across the curriculum have been proven in its implementation. However we feel there was still significant advanced content that would benefit undergraduates and help reduce the cyber-security professional deficit…In fact we illustrate there are some topics that are simply not found in any other discipline. (p. 116)

Rowe, Lunt, and Ekstrom (2011) define the following specialization classes: Cyber threats and penetration testing; cyber defense and systems administration; and cyber response and forensics.

In another viewpoint, Meiselwitz (2008) defines a single class that incorporates numerous IA topics, such as: threats, forensics, social and legal issues. This compression into a single class was too aggressive. It was highly unlikely a student will remember the concepts presented, let alone comprehend the concepts at any level of mastery. Some of these concepts like: forensics; threats and penetration testing; and defense and system administration need to be addressed in detail. The problem was which classes belong in IT and which classes belong in IA.

According to the ACM and IEEE (2005), there are several pervasive themes in IT, such as: system integration; use of abstraction; user advocacy; information assurance; adaptability; and professionalism. Applying these themes will help define IA concepts that should be addressed directly in IT. System integration lends nicely to concepts around system defenses, like system hardening and system penetration testing (Dimkov, Pieters, & Hartel, 2011). IA lends nicely to a few different topics, such as: business continuity and disaster recovery (Ward et al., 2009); and system threats and incident handling (NIST, 2008). Finally user advocacy and professionalism lends nicely to defining privacy and ethics (Fass, 2008).

Very little research has been completed on what IA concepts should be integrated throughout IT. Some older research completed by Null (2004) suggested integrating concepts like: security risks and threat sources (Clark, Singleton, Tyree, & Hale, 2008; Foley 2009); spoofing; reconnaissance software; encryption; operating systems vulnerabilities; denial of service; malware; remote monitoring; secure email; firewalls; and mobile code. This is somewhat supported by research completed by Dark, Ekstrom, and Lunt (2005). Dark, Ekstrom, and Lunt (2005) base their work on seminal research

completed by Maconachy, Schou, Ragsdale, and Welch (2001). Dark, Ekstrom, and Lunt's (2005) research was somewhat dated now, but does define eleven core IA competencies. They are: fundamental aspects; security mechanisms and countermeasures; operational issues; policy; attacks; security domains; forensics; information states; security services; threat analysis models; and vulnerabilities.

These eleven concepts defined by Dark, Ekstrom, and Lunt (2005) do follow the criteria of IT defined by ABET (2012). ABET (2012) defines the core fundamentals of IT as the following: human computer interaction; information management; programming; networking, web systems and technologies; IA and Security; system administration and maintenance; and system integration and architecture. The core fundamentals defined by ABET (2012) follow the recommendations defined by ACM and IEEE (2008b). The ACM and IEEE (2008) define IT as the following knowledge areas: IT fundamentals; human computer interaction; information assurance and security; information management; integrative programming and technologies; math and statistics for IT; networking; programming fundamentals; platform technologies; system administration and maintenance; system integration and architecture; social and professional issues; web systems and technologies. Many of these concepts can be applied within IT, but do advanced classes like forensics belong in an IT program, or should they be taught in a specialized IA program.

Current State of Information Assurance

Information assurance is the newest curriculum within the technology programs. Much of the research presented on IA curriculum development was around the creation of only a few specific security classes. Typically these specialty classes are around

computer forensics (Schwarz, 2005). Needless to say, the methodologies curricula of IA applied to existing programs are difficult at best, but applying the concepts of IA to an entirely new program was extremely difficult. Goel et al. (2006) clarify this difficulty by stating, "Because IA was a complex subject that spans several disciplines, it was important to build a rich context so that concepts can be assimilated rapidly across multiple disciplines" (p. 3). This is one of the driving issues within IA. The core concepts of IA span multiple disciplines.

Another factor affecting the creation of all IA programs are the rapidly changing landscape of vulnerabilities. As Goel et al. (2006) state, "A fundamental problem with IA curricula, especially with hands-on exercises, was that the curriculum materials quickly become obsolete as threats and vulnerabilities evolve rapidly" (p. 7). This was a huge problem for any college and university. Unlike a typical lifecycle that tries to refresh technology, including hardware every three years, security vulnerabilities are constantly changing requiring new skill sets and tools to be created constantly.

The problem with IA as a stand-alone program was the lack of industry standards. Neither the ACM nor the IEEE Computer Society has developed any recommendations or guidelines for developing an IA program. ABET has also not identified any IA requirements for either existing technical programs or for a stand-alone IA program (Schweitzer, Humphries, & Baird, 2006). The US government was the only entity that has established a criterion for an IA program. The National Security Agency (NSA) and the Department of Homeland Security (DHS) have developed a program known as Center of Academic Excellence (CAE) for IA.

The NSA (2009) defines the following criteria for a CAE:

1. Partnership in IA education with minority colleges and universities.

2. Evidence that IA was not treated as a separate discipline, but as a multidisciplinary science.

3. The academic program demonstrates how the university encourages the practice of IA.

4. The academic program encourages research in IA.

5. The IA curriculum reaches beyond the normal geographic borders of the university.

6. Faculty was active in current IA practice and research, and contributes to IA literature.

7. The university library or the IA Center maintains state-of-the-art IA resources.

8. The academic program has declared IA concentrations.

9. The university has a declared center for IA education or a center for IA research from which IA curriculum was emerging.

10. University IA faculty consists of more than one individual devoted full time to IA. (p. 1)

For specifics on IA curriculum, the NSA and DHS base their requirements on the National Training Standard for INFOSEC Professionals standards developed by the Committee on National Security Systems (CNSS). The CNSS define six different standards: National Training Standard for Information Systems Security (INFOSEC) Professionals (NSTISSI-4011); National IA Training Standard for Senior System Managers (CNSSI-4012); National IA Training Standard for System Administrators

(CNSSI-4013); IA Training Standard for Information System Security Officers; National Training Standard for System Certifiers (NSTISSI-4015); and National IA Training Standard for Risks Analysts (CNSSI-4016).

The problem with many of these standards was they are old and have not been recently updated. The NSTISS (1994) does define two distinct levels of knowledge, which can easily be applied to existing programs versus a new stand-alone IA program. The NSTISS (1994) defines the following two levels:

> Awareness Level. Creates sensitivity to the threats and vulnerabilities of national security information systems, and recognition of the need to protect data, information and the means of principles and practices in INFOSEC.
>
> Performance Level. Provide the employee with the skill or ability to design, execute, or evaluate agency INFOSEC security procedures and practices. This level of understanding will ensure that employees are able to apply security concepts while performing their tasks. (p. 5)

These definitions are the perfect analogy of why it is important to have existing programs and new IA programs incorporate curriculum of IA. No matter the program a student was enrolled in, a basic awareness level of IA concepts is a requirement. Without a basic level of awareness, students will propagate common security vulnerabilities (Meiselwitz, 2008). These common vulnerabilities are critical to secure systems and need to be addressed.

Experts within IA are also required to help promote, engineer, and architect secure systems (Meiselwitz, 2008; Goel et al., 2006). Specialist in security architecture, computer forensics, business continuity, and governance regulation and compliance

(GRC), are required for businesses not only to be successful, but also secure and in compliance with the numerous required regulations that need to be followed

Evaluation and Summary

Information Assurance through Repetition and Mastery

As observed from the research presented, several core concepts are repeated over and over again. Appending an IA class to the end of a CS, SE, or IT program was not an ideal inclusion of IA concepts within a technical program. Learning was a process of re-enforcement (Knewstubb & Bond, 2009).

Many times concepts are repeated to re-enforce a key concept. The skill of drawing was being practiced over and over again, but the context of the drawing may change. Another example was a music student learning a piece of music. The first time the student reads a sheet of music there will be numerous errors. The student then practices over and over again until the song was perfected. A final example is a medical student perfecting a surgical procedure. A student is not a master, nor should they be considered a master after an initial surgery. A medical student practices a procedure over and over again before they are allowed to perform an operation by themselves. Knewstubb and Bond (2009) support the concept of learning through repetition, as they state, "Teaching in the arts is about having ideas to evaluate and, secondly, that repetition was important for successful teaching and learning" (p. 190). The understanding and application of IA is an art form, much like learning or mastering a fine motor skill or like drawing or a surgical procedure. The act of learning / practicing should be repeated, but the context of that learning needs to be changed. Similarly the concepts of IA need to be

repeated, but the context needs to be changed. In other words, IA concepts need to be repeated throughout different classes that define different contexts.

The idea of learning through repetition is further supported by Papadopoulos, Demetriadis, Stamelos, and Tsoukalas (2009), who support the idea of not lumping concepts within a single class. Research by Papadopoulos, Demetriadis, Stamelos, and Tsoukalas (2009) supports the idea that learning is enhanced by repetition, specifically when the repetition was spaced. Papadopoulos, Demetriadis, Stamelos, and Tsoukalas (2009) determined that memory performance was enhanced when learning repetitions were spaced rather than massed together.

Enhancing memory performance through repetition is further supported by Greene's (1989) research. As Greene (1989) states:

> When an item was presented on a list, the subject must decide how much study to devote to it. This decision was based in part upon how easy the subject believes the item will be to remember later. If an item was repeated in massed fashion, it will seem very familiar to the subject on its second occurrence. The subject may mistakenly believe that the item was already well learned and may devote little additional rehearsal to it. As a result, the subject allots less rehearsal time and processing resources to the second occurrence of massed items than to once-presented items. As spacing between repetitions increases, familiarity of a repeated item decreases, leading to an increase in rehearsal allotted to repetitions and an increase in recall. (p. 371)

This research is specific to items within a list, but the concepts can be applied to overall curriculum that needs to be applied across an entire program. To have IA concepts

massed at the end of a program was not an effective method of increasing memory retention of IA concepts and methodologies.

The concept of repetition also is applied from a business perspective. Artner (2001) argues that IT managers need to apply the repetition imperative. Basically Artner (2001) defines repetition imperative as the ability of a technical manager to say what needs to be said, and not to be afraid to say it over and over again. Artner (2001) argues that repetition is not because people are stupid, but exactly the opposite. Repetition was often used to motivate intelligent people and keep them on target. People naturally migrate to tasks they enjoy and need to be reminded of tasks that may not be as enjoyable. Artner (2001) defines three reasons for the need of repetition. The first reason was to emphasize the important of an idea or concept. The second reason was to re-iterate complex instructions. The final reason for repetition was to protect you by providing proper control.

The importance of repetition is easily transferred back and forth between business and academia. Good technical managers motivate and keep resources on task by use repetition. Good instructors repeat core ideas and concepts to emphasize statements of importance. It is important that repetition is applied across an entire program to emphasize important concepts, in this case concepts related to IA. Businesses are customers of the academic system and therefore drive program requirements. Businesses routinely use repetition to emphasize goals. This can easily be transferred back to academia where the goal was now a core concept that needs to be repeated.

Seminal researcher on learning, Attewell (1992) presents another viewpoint that examines the incorporation of new technologies within a business. Often this diffusion of

technology within a business is complex and time consuming. Expertise in the new technology may not exist within the organization. Attewell (1992) defined this lack of experience as a knowledge barrier. Unlike consultants that specialize in the new technology being implemented, a business does not have the expertise that was acquired by learning the technology through repetition. A consultant becomes skilled by learning through repetitive implementations. A business implementing this new technology does not have this luxury or experience. Attewell (1992) states, "The long-term solution to this demand for expertise was to automate and to standardize" (p. 16). Even though Attewell (1992) was referencing expertise of a specific technology within a business, the same idea can be applied to IA and education. A standardized method of teaching IA curriculum within technical programs needs to be established. Clearly there was a need to repeat IA concepts throughout a CS, SE, or IT program and not append a class at the end of the program.

In addition to repetition, mastery of IA concepts can be achieved through applying the concepts of Bloom's taxonomy. A great deal of research related to Bloom's Taxonomy (1956) and mastery of CS, SE, IT, and IA has been completed (Cooper, Finley, & Kaskenpalo, 2010; ACM & IEEE, 2004; ACM & IEEE, 2008b; Cooper et al., 2010; Reynolds & Goda, 2007; Gray, St. Clair, James, & Mead, 2007; Goel et al., 2006; Kamali, Liles, Winer, Jiang, & Nicolai, 2005; Conklin & Dietrich, 2007; Hjelmas & Wolthusen, 2006; Livermore, Baker, Krolczyk, & Saurbier, 2011; Whitman & Mattord, 2005; Bhagyavati, Naugler, & Frank, 2005). Proficiency was built upon the repetition of concepts and building on the lower levels of Bloom's Taxonomy (1956) of knowledge,

comprehension, and application and being able to apply and analyze those concepts (Reynolds & Goda, 2007; Cooper et al, 2010).

The Need for Business Collaboration within Information Assurance

It is critical that academia works with both the public and private sector to develop programs that are current and relevant (Benamati, Ozdemir, & Smith, 2010; Abernethy, 2011; Reid & Gilbert, 2007). Mead and Jarzombek (2010) iterate this same concern. Collaboration was critical for both academia and the private and public sectors to be successful in advancing IA. Mead and Jarzombek (2010) state, "To achieve these goals, the Software Assurance program needs the active participation of practitioners involved in every stage of the Software Development Life Cycle (SDLC) and from all sectors of the economy, including software project managers, professionals working in acquisition, and members of relevant standards bodies" (p. 29).

Unfortunately the need to collaborate may be a paradox in itself. Mead and Jarzombek (2010) explain, "Few practitioners today have the background needed to build secure software" (p. 27). According to Mead and Jarzombek (2010), software assurance currently was dispersed among several dissimilar programs, such as CS, SE, and IT. Yet at the same time, private and public organizations have concerns about vendors and contractors delivering secure software. Regrettably, this issue was compounded by the fact the private and public organizations may not be leading by example, which was the next problem that needs to be addressed.

Clearly all organizations need to show a business justification for IA. According to Mead and Jarzombek (2010) this was easily demonstrated by showing the cost of

integrating IA concepts throughout a process, in this case the SDLC, and the cost of fixing issues at the end of a process. There is a clear cost savings by incorporating IA concepts from the beginning and throughout, than applying the same concepts at the end of a process and repairing any issues that may have been discovered.

The need to lead by example is further supported by Bishop and Frincke (2008). According to Bishop and Frincke (2008), industry and government must encourage the growth of IA in their own actions. Bishop and Frincke (2008) further state, "Leading by example conveys to students that computer security was a discipline" (p. 55). Students are requesting skills from academic institutions that are being asked for by organizations. If organizations are not asking for graduates with IA experience, then students will not ask to be taught these skill sets. On the other hand if a student knows that many organizations are looking for a skill set in IA, they know having that skill set will make them more attractive to an organization than a student with no IA skill sets.

This collaboration between academia and the public and private sectors seems simple at first glance, but when looked at in detail there are some issues that need to be addressed. According to Mead and Frincke (2008), academia has different constraints than many of the private and public sectors. For example, in a private or public sector setting a manager or director will make a decision and communicate that decision to subordinates to implement. The subordinates are expected to carry out this decision, but in an academic setting, this decision was somewhat up for interpretation.

Unfortunately the field of IA is rapidly developing. Approval of new classes takes time within in college or university. This lends evidence to the need to integrate IA concepts within existing classes and not appending them to the end of a program. It was

easier to incorporate current events within IA into existing classes than developing a whole new class to review current IA events.

For example research by Peltsverger and Teat (2009) support the need to incorporate current IA events into existing classes. Peltsverger and Teat (2009) show that incorporating small discussion groups or requiring research papers on current events can achieve adding current events, in this case to IA classes. According to Peltsverger and Teat (2009), by incorporating small discussion groups and research papers on current events within IA, this causes students to be excited about learning IA concepts.

Evaluation of the Current Workforce

A recent survey performed by Lee, Bagchi-Sen, Rao, and Upadhyaya (2010) gives a glimpse of the current anatomy of the IA workforce. Unfortunately the research was somewhat limited due to the sampling size, but the research does show some interesting trends between IA and IT and the workforce age.

The first interesting finding was the perceived skill sets required for IA professionals versus IT professionals. The results of Lee, Bagchi-Sen, Rao, and Upadhyaya (2010) survey showed a need for a higher level of knowledge in IT and overall business skills within IA, compared to their counterparts within IT. This may not be that surprising, since IA professionals need to work closely with the business to determine the level of risk that was acceptable to the business.

The level of the workforce also showed some statistical differences between IA and IT. For example the level of knowledge within business increased for both IA and IT managers, but the need to retain technical skills was different. Lee, Bagchi-Sen, Rao, and

Upadhyaya (2010) survey showed IA managers retained their technical skills, while their counterparts in IT had a relatively low need to retain technical skill sets.

Finally Lee, Bagchi-Sen, Rao, and Upadhyaya (2010) define the following IA skill sets: risk assessment; policy and guideline development; business continuity plan development; access control implementation; user and event monitoring; cryptography; security equipment and software management; physical security; secure software design and development; business operations recovery; incident investigation; litigation and prosecution; user education and training; research and development, vendor and customer relationship management; intelligence and information collection; and finally security policy enforcement (Baird & Gamble, 2010). Lee, Bagchi-Sen, Rao, and Upadhyaya (2010) also were able to define four distinct groups from the sampling: front-liners, directors, developers, and soldiers in the trench. These distinct groups are a good simplification of the different groups defined by the CNSS.

As one would suspect, each of these groups concentrate on different skill sets. According to Lee, Bagchi-Sen, Rao, and Upadhyaya (2010), the front-liners focus on the following: designing and implementing access controls; monitoring user access and security events; managing security hardware and software; business continuity; compliance with security policy; and interactions with customers and vendors. Directors according to Lee, Bagchi-Sen, Rao, and Upadhyaya (2010) are more responsible for managing the developers and soldiers in the trenches, but are less directly involved with tasks performed by the front-liners. Developers are a little more unique, in that they focus on research and development and less on operational aspects. According to Lee, Bagchi-Sen, Rao, and Upadhyaya (2010), developers are a little broader in that they include

software developers and security awareness and training. Finally the soldiers in the trenches are dedicated to operational aspects of IA, such as: event monitoring; access control; incident investigation; and policy enforcement (Baird & Gamble, 2010). According to Lee, Bagchi-Sen, Rao, and Upadhyaya (2010), this group tends to have less interpersonal communication skills and deal less with system planning and design.

The final finding Lee, Bagchi-Sen, Rao, and Upadhyaya (2010) discovered was the difference in education levels between different age groups. Lee, Bagchi-Sen, Rao, and Upadhyaya (2010) defined three different age groups: new generation, mid generation, and first generation. New generation IA workers tended to have a higher number of bachelor degrees compared to the other two defined age groups. The mid generation had fewer bachelor degrees than the other two defined age groups and finally the first generation had a higher than normal level of graduate degrees compared to the other two defined age groups. CS was the preferred degree found in a majority of the people surveyed in the new and mid generations, while an MIS degree was held by a majority of the workforce considered to be in the first generation. The last interesting discovery was the inclusion of IA as a degree. The survey sample did include people with IA degrees. The few that were documented were found in the new generation, but also were found in the first generation group as well. Lee, Bagchi-Sen, Rao, and Upadhyaya (2010) make an assumption that first generation workers typically go back to school between the ages of 38 and 45, which could explain the discovery of new programs like IA in this age group.

Unfortunately information around the anatomy of the IA workforce was limited. In a second study, Frost and Sullivan (2011) surveyed over 10,000 IA professionals, but

this study focused more on demographics of the IA population. Frost and Sullivan (2011) did capture some data around future trends. Frost and Sullivan's (2011) survey did gather data from 10,413 subjects. The survey defined the following as top security threats (in order of importance): application vulnerabilities; mobile devices; viruses and worm attacks; internal employees; hackers; contractors; cyber terrorism, cloud-based services, and organized crime. In the same study Frost and Sullivan (2011) also captured what IA professionals are requesting to be trained. They are the following (in order of importance): information risk management; system development security; forensics; end-user security training; security architecture and models; access control systems and methodology; security management practices; and business continuity and disaster recovery plan. These discoveries seem to align somewhat with the SIGITE, $(ISC)^2$, and the CNSS.

Summary

There is a clear industry and academic need for IA to be both an integrated approach within existing programs and new IA specialty programs. According to Rao, Gupta, and Upadhyaya (2007), IA is driven by three core business requirements; they are compliance, protection of shareholder value, and cost. These business requirements are clearly discussed numerous times throughout the literature. For example, several regulations are referenced along with several different references on cost benefit analysis.

Brewer (2005) points out the affects of regulation by specifically analyzing the business implications of Sarbanes and Oxley. Brewer (2005) states, "Security concerns alone render sufficient reason for using a formal architecture design process and a highly disciplined approach to protect information technology systems, networks, and

applications" (p. 73). This clearly indicates that all technical graduates need to have a high degree of discipline within IA. Brewer (2005) goes on to state, "Reactionary approaches to security design place the hackers at an advantage and the IT staff in a no-win downward spiral of iterations of discovery, patch, and run" (p. 73). Blyth and Kovacich (2001) support both the protection of shareholder value and the associated costs of implementing an IA program. For example Blyth and Kovacich (2001) state that one driver for IA is the ever-changing laws and regulations associated with it. Blyth and Kovacich (2001) rationalize the need for IA within an organization in order to protect the shareholder value and keep a competitive advantage. Contracts are another huge factor that can drive the adaptation of IA curricula (Blyth & Kovacich, 2001). Finally Blyth and Kovacich (2001) state that many organizations have a desire to protect their information from unauthorized access, but may also be driven by domestic and international laws and regulations to also enforce this protection. Gilbert (2009) echoes many of the same factors stated by Blyth and Kovacich (2001). Gilbert (2009) also adds disgruntled employees, increased litigation, and mergers and acquisitions as other factors that are driving the need for IA curriculum.

Fitzgerald (2008) points out another important business driver around protecting shareholder value. Many organizations will at some point experience downsizing, mergers, acquisitions, and the need to trust business partners. None of these scenarios can have a cookie cutter textbook solution applied to them, but it was the responsibility of IA experts to provide a means to securely execute any of these scenarios. Undergraduate students need to understand the fundamental concepts and how to apply them to real-

world scenarios. Having a single class at the end of a program does not give the exposure needed to succeed.

Several recent research papers within IA also support the use of a qualitative research methodology (Bryielsson, 2009; Cannoy & Salam, 2010; Zambon, Bolzoni, Etalle, & Salvato, 2007; Dottore, 2009; Spears & Barki, 2010; Botta et al., 2007; Mead & Hough, 2006; Miller & Dettori, 2008; Werlinger, Hawkey, & Beznosov, 2008). Werlinger, Hawkey, and Beznosov (2008) observed 22 security practitioners to determine their activities and interactions with stakeholders. This research proposed to interview several security practitioners as stakeholders within academia. Research by Miller and Dettori (2008); and Brynielsson (2009) further supports the use of qualitative interviews to capture perceptions of employer's on IT learning outcomes, which was the objective of this research. The use of case studies was supported by the research of Cannoy and Salam (2010) and Dottore (2009).

CHAPTER 3. METHODOLOGY

This chapter describes the methodologies that were deployed to examine the inclusion of information assurance curriculum within existing undergraduate computer science (CS), software engineering (SE), and information technology (IT) curricula. The chapter is first broken down into describing the purpose of this research. The second section of this chapter describes the research design. The third section describes the sample population and sampling techniques that were deployed. The fourth section describes the instrumentation and measures. The fifth section discusses the methods implemented for data collection. The sixth section describes the processes used when analyzing the collected data analysis. The final section examined the validity and reliability of the collected data and research.

The problem this study examined was the lack of IA topics being taught within existing computer science (CS), software engineering (SE), and information technology (IT) curricula (Myers & Riela, 2008). The purpose of this research was to determine what IA topics should be taught within CS, SE, and IT curricula. Findings from this research may assist university administrators to better incorporate IA curricula into existing CS, SE, and IT programs and curriculum.

This research was significant because there was a deficiency of research available on what IA concepts should be integrated within CS, SE, or IT. Numerous studies (Petrova, Kaskenpalo, Philpott, & Buchan, 2004; Bogolea & Wijekumar, 2004; Taylor & Azadegan, 2006; Wang, 2008) had been compiled regarding the need to integrate IA curricula into existing CS, SE, and IT programs, but little research had been compiled on what IA concepts specifically needed to be incorporated into existing CS, SE, and IT

coursework and programs. This research evaluated what IA concepts should be integrated into CS, SE, and IT and if these concepts should be repetitively applied throughout a program (Papadopoulos, Demetriadis, Stamelos, & Tsoukalas, 2009).

The review of literature showed the need to integrate IA curriculum within CS (Cooper et al., 2010; Crowley, 2007; Dark, Ekstrom, & Lunt, 2005; Dimkov, Pieters, & Hartel, 2011; Geoghegan, 2008; Hjelmas & Wolthusen, 2006); the need to integrate IA curriculum within SE (Lester & Jamerson, 2008; Lester & Jamerson, 2009; Mead & Hough, 2006; Conklin & Dietrich, 2007; Maqsood & Javed, 2007); and the need to integrate IA curriculum within IT (Rowe, Lunt, & Ekstrom, 2011; Dimkov, Pieters, & Hartel, 2011; Null, 2004; Dark, Ekstrom, & Lunt, 2005; Maconachy, Schou, Ragsdale, & Welch, 2001). What the literature did not distinctly show was what specific IA concepts should be taught within CS, SE, or IT. Most of the current literature that focuses on IA concepts (ACM & IEEE, 2008b; Tipton, 2009) is now several years old and does not address current concerns, such as cloud-computing. Another possible issue was that current Information Security leaders may not understand what IA concepts exist today.

Research Design

The purpose of this qualitative case study was to determine what IA topics should be taught within CS, SE, and IT curricula. Several other researchers (Bryielsson, 2009; Cannoy & Salam, 2010; Zambon, Bolzoni, Etalle, & Salvato, 2007; Dottore, 2009; Spears & Barki, 2010; Botta et al., 2007; Mead & Hough, 2006; Miller & Dettori, 2008; Werlinger, Hawkey, & Beznosov, 2008) support the use of qualitative analysis. Specifically this study utilized a case study methodology using multiple cases to evaluate the current perceptions of IA practitioners and then synthesized a curriculum and

compared the proposed curricula against current professional and academic recommendations. Qualitative research methods are used to study human behavior and behavior change (NIH, 2011; Creswell, 2009; Creswell, 1998).

The use of multiple case studies was supported by Yin (2009). According to Yin (2009), each participant might be considered an individual case study, but the research as a whole would cover multiple participants or in this case multiple cases. Each interviewed practitioner was an individual case, but a compilation of cases was required to fully cover the breadth of this research. Yin (2009) defined five components that should be present in all case studies: a study's question; its proposition, if any; its unit of analysis; the logic linking the data to the Hypotheses; and the criteria for interpreting the findings.

This research addressed the following question:

Research question

What information assurance core competencies should be included in computer science, software engineering, and information technology curricula?

According to Yin (2009) a case study typically addresses how or why questions. This research was focused on what IA concepts should be included, but there are several propositions on how this should be completed.

Proposition

The literature review helped define three different propositions, one for each areas of study: CS, SE, and IT.

Proposition One: Computer Science

The literature review defined six core IA classes that should be included within CS and two other optional classes. The classes defined by the literature review are: cryptography, forensics, network security, ethics, incident handling, and security architecture; with risk management and privacy as optional courses (Cate, 2009). The first proposition was to test was if IA practitioners agree these are core IA concepts that should be included in CS, or should be solely included within an independent IA curriculum.

Proposition Two: Software Engineering

The literature review defined several IA concepts that should be addressed throughout the SE curriculum. They are: buffer overflows; SQL injection attacks; cross-site scripting; cross-site request forgery; click jacking of code; broken session authentication and session management; insecure direct object references; security misconfigurations; insecure cryptographic storage; failure to restrict URL access; insufficient transport layer protection; and invalidated redirects and forwards. The literature review also defined two core IA concepts that should be included in SE. They are: cryptography and software risk and project management. Proposition two tested if IA practitioners agree with incorporating common software vulnerabilities throughout a SE curriculum and appending two fundamental IA classes within the SE curriculum.

Proposition Three: Information Technology

The literature review defined several core IA concepts that are combined to create a set of classes specific to IA within IT. These core IA concepts are: fundamental aspects; security mechanisms and countermeasures; operational issues; policy; attacks; security domains; forensics; information states; security services; threat analysis models; and vulnerabilities. Proposition three validated if IA practitioners agree that these core IA concepts should be included within IT, or if any of these concepts should be taught in a dedicated IA curriculum.

The unit of analysis goes back to the proposed research question: What IA core competencies should be included in CS, SE, and IT curricula. The literature review suggested certain integration methodologies and specific IA topics that should be covered (Dark, Ekstrom, & Lunt, 2005; Lester & Jamerson, 2009; Null, 2004). The literature review outcomes needed to be validated against real world scenarios. To measure the validity of these outcomes, this research interviewed several IA practitioners that are in a leadership positions. A leadership role guaranteed some level of knowledge in CS, SE, or IT. By selecting a development manager or system manager that limited the input for all three defined areas of study: CS, SE, and IT.

This research utilized a pattern-matching model to analyze the data collected from the multiple cases that were observed (Yin, 2009). The hope was that by using a pattern-matching analysis method this study would be able to draw a conclusion if the Hypotheses defined from the literature review matched the perceptions of IA practitioners.

Because of the qualitative nature of this study it was impossible to find any statistical significance, as a researcher would be able to calculate in a quantitative study (Yin, 2009). According to Yin (2009) an important alternative strategy was "to identify and address rival explanations" (p. 34). To address this concern this research utilized a replication of methodology between the different cases. Different IA leaders were selected from different organizational demographics. There was a possibility the different business verticals and different size organizations may influence the data collected from the associated IA leaders. Another rival explanation may be the level of education completed by the IA leader.

Population and Sample

The population of this research was limited to Information Security experts who either would directly hire college graduates for information assurance positions within their organizations; or who oversee information security / information assurance departments or are directly impacted by the IA skill sets of their subordinates. Various business verticals were also selected to validate if these were common perceptions throughout information assurance. These business verticals included:

- A medical device manufacturer
- A third-party logistics brokerage
- Information Security Consulting
- Healthcare
- Information security value add reseller (VAR)
- Accounting and Auditing
- Specialized healthcare

- Computer software
- Media and Information

This population was selected because these experts see the current threat landscape and knows the gaps in the skill sets of their current employees. All participants were directly responsible for the hiring and termination of IA professionals within their respected organizations (see Appendix B).

According to Creswell (1998, 2009), a case study should utilize a defined generalized group. The National Institute of Health (NIH) (2011) also supports the use of a generalized group for replication. The sampling must support both the support of this research and any rivals that may be associated to this research. The generalized group will be selected from IA experts who have worked in the field several years and directly manage IA resources. The IA experts that were selected had a broad enough level of experience to successfully answer what IA skill sets are missing within CS, SE, and IT resources in their own organizations. It was possible the size of the organization, the business vertical, and the level of education completed by the IA leader may have had some influence on the required IA skill set. Because of this concern, each replication utilized a different business vertical and organizational size. Education level of the IA leader was unknown until the case data was collected.

The initial sampling population consisted of ten different cases (see Appendix B). According to Yin (2009), when using multiple cases "a sampling logic should not be used, the typical criteria regarding sample size also are irrelevant" (p. 58). In the case of using multiple case studies a researcher needs to consider both literal and theoretical replications. Ten literal replications were selected based on Yin's (2009) discussion on

detecting a study's effect. According to Yin (2009) if a theory was straightforward a researcher could use two or three literal replications; on the other hand if the theory was subtle or there was a need for a high level of certainty, five or more replications should be used. Eisenhardt (1998) and Stake (2005) also supports the use of multiple cases.

For the number of theoretical replications, Yin (2009) stated it was important to consider the sense of importance for rival explanations. The more rivals the more replications that are required. This research utilized ten initial theoretical replications. Ten replications were sufficient to fully explore any rival explanations that were discovered in this research.

Setting

The setting for this research was completed within the Midwest region of the United States of America. Twenty initial cases were randomly selected for the initial population. Ten cases were selected and analyzed; with the other ten cases reserved in case any of the initial cases needed to back out, or in case more cases were needed. The initial ten cases included the following: Case one is a Senior Security Manager of a large medical device manufacturer; case two is a Senior Security Manager for a large global third-party logistics brokerage; case three is a Security Director of a brand loyalty provider; case four is a Security and Compliance Director of a large healthcare provider; case five is the Managing Director / founder of a medium sized security consulting firm; case six is a Security Manager of Risk Advisory Services for a large CPA firm; case seven is a Chief Information Security Officer for a specialty healthcare system; case eight is the Chief Technology Officer / Chief Information Security Officer of a compliance software manufacturer; case nine is the Director of Information Assurance for a large

global publisher; the final case is an independent Information Security consultant. Half of the cases selected are also adjunct faculty instructors or professors at various regional colleges / universities and have influence on the included content within the program(s) they teach in (see Appendix B).

The information assurance leaders were selected randomly from different contacts obtained by the researcher. The first participant's email and phone number were obtained through a local chapter meeting related to the Information Systems Security Association (ISSA). The second participant's email and phone number were obtained through a previous employer. The original request was to the original security manager at the previous employer, but the researcher was informed that the old security manager was promoted and not directly responsible for Information Security. The contact information to the new security manager was provided, where initial contact was made through an email. The third participant's email and phone number were obtained through a local chapter meeting related to the ISSA. The forth participant's contact information was obtained through recurring meetings of the Minnesota Health Information Exchange collective. The fifth participant's email and phone number were acquired through an introduction at a local security conference known as Secure360. The sixth participant's contact information was obtained through a local chapter meeting related to the ISSA. The seventh participant's email and phone number were obtained through a bi-monthly meeting of the Healthcare Security Professional Interest Group (HSPIG). The eighth participant's contact information was obtained through a local meeting related to the International Information Systems Security Certification Consortium (ISC2). The ninth participant's email and phone number were acquired through an introduction at a local

university where both the researcher and participant were adjunct instructors. The final participant's contact information was obtained through a SANS community class. An initial communication was emailed to each participant asking for his or her participation in a case study. A follow up interview was completed in person. Each participant was asked to sign a letter of consent. The identities of the participants were kept anonymous in the results of this research.

Field Test

Five Information Security experts were selected to field test the questionnaire. These five participants were used to validate the questionnaire and to provide critical feedback on usability. One suggestion that was made by the participants was to include an open question at the end of the interview to capture any additional thoughts or comments that were not captured throughout the interview, which was included in the final interview instrument. The sampling used for the field test was not used for the actual data collection or included in the results of the study.

Instrumentation

Participants were interviewed using a semi-structured interview instrument (Appendix A). This was the ideal instrument to gather information, since the expected answered will be short, and there are no predefined answers. According to the NIH (2011) and Creswell (1998, 2009), this was the appropriate method to use if the researcher is expecting short and undefined answers. The use of a focused interview was also supported by Yin (2011). According to Yin (2011), focused interviews are ideal for corroborate certain facts that have been established. In the case of this research, the Hypotheses were tested.

The interview instrument consisted of 25 questions and was broken down into three themes: one theme for CS, a second theme for SE, and the final theme for IT. From the literature review several sub-themes were established.

CS was broken down into eight sub-themes. They are the following:

1. Cryptography
2. Forensics
3. Network Security
4. Ethics
5. Incident handling
6. Security architecture
7. Risk management
8. Privacy

SE was broken down into three sub-themes. They are the following:

1. Common vulnerabilities (code injection, buffer overflows, cross-site scripting, etc).
2. Cryptography
3. Software risk and project management

IT was broken down into eleven sub-themes. They are the following:

1. Fundamental aspects of security
2. Security mechanisms and countermeasures
3. Operational security
4. Policy creation and management
5. Attacks

6. Security domains
7. Forensics
8. Information states
9. Security services
10. Threat analysis
11. Security vulnerabilities

The first question in the questionnaire was a demographic question used to gather the participant's title and role within his or her organization. The assumption was that the position title and role would reflect the participant's normal role and responsibilities within the organization. This helped validate the credibility of the participant's contribution to the study. The demographic data collected provided evidence that the participants are currently leaders within Information Security or IA.

The second demographic question was to establish the level of education completed by the participant. This demographic was collected to validate if the level of education may have an influence on the rest of the data that was collected. The assumption was the level of education would not have an influence on how the rest of the questionnaire was completed.

The final set of questions was broken down into three core themes of CS, SE, and IT. The themed questions were designed to capture the perception of the participant of a particular sub-theme and document why they had that perception. The sub-themes were coded through evaluations of concept and word repetition within the literature review. A final open-ended question was asked on any other comments the participant wished to present.

Data Collection

According to Yin (2009), there are three principles of data collection. They are: Use multiple sources of evidence; create a case study database; and maintain a chain of evidence. By following each of these principles for data collection, this helped establish construct validity and reliability (Yin, 2009). Each of these principles is elaborated in detail.

The use of multiple sources of evidence was critical for this study. Multiple case studies were collected and evaluated, along with an extensive literature review. An extensive literature review was completed first to establish a proposition to test within the multiple case studies that were evaluated. The literature review evaluated current research on the integration of IA curriculum in CS, SE, and IT. Different integration methodologies were evaluated. From the literature review, a list of hypothesized topics was established that should be integrated within CS, SE, and IT.

The hypothesized topics for CS, SE, and IT were then validated against ten Information Security experts, five of which are also practicing adjunct instructors / professors. The perceptions of the ten selected Information Security Experts were captured when asked about their opinions on including the hypothesized topics within CS, SE, and IT. An initial pool of twenty Information Security experts was selected. From the initial twenty Information Security experts selected, only ten were interviewed. The other ten cases were used for reserve case studies in case any of the participants were not available or did wish to participate. Two of the initial cases were replaced due to unavailability.

The use of multiple case studies and an extensive literature review allowed this research to evaluate a broader range of historical and behavioral issues (Yin, 2009). According to Yin (2009), by including multiple source of evidence this will allow this research to develop a converging line of inquiry. This converging line of inquiry was the basis of triangulation. This research utilized a data triangulation, by converging the evidence collected in the semi-structured interviews and the evidence collected in the literature review. By using multiple evidence sources and data triangulation, the research was able to corroborate the proposed research question. Data triangulation addressed potential problems of construct validity when using case studies by using multiple sources to measure the same phenomenon (Yin, 2009).

In order to establish reliability, the evidence collected was imported into a database where it was organized, archived, and analyzed. According to Yin (2009) the lack of a formal database was a major shortcoming in many case studies. This research collected interview information by audio recording all interviews the manually transcribing the interview. All participants agreed to be recorded, so two artifacts were preserved; the audio recording and an electronic transcription of that recording. All literature reviewed was documented and saved. This research stored all case documents as a Microsoft Word format or Adobe PDF document format. NVivo v10 was utilized as the case management software. All collected audio and written evidence was stored and analyzed within NVivo v10.

Further reliability was established by maintaining a chain of evidence (Yin, 2009). According to Yin (2009), any case study research that formally establishes a chain of evidence should allow the reader of that case study to trace any evidence from the

research question to the conclusion and vice versa. By maintaining a formal chain of evidence, construct validity was addressed and therefore also increases the overall quality of the research (Yin, 2009).

Data Analysis

This research deployed the data analysis method for qualitative analysis as defined by Creswell (1998). The first step was to organize and prepare the collected data for analysis. First the collected interview data was transcribed for encoding the data into detected themes and patterns (Yin, 2009). Second, all field notes were transcribed and correlated back to the interview data (Creswell, 1998).

The second step was to establish a general sense of the collected data and try to determine an overall meaning. Field notes were compared along with the transcribed interview data to validate any general senses that may be observed (Creswell, 1998).

Step three was to encode the collected data into segments. These segments helped drive any emerging themes and descriptions from the collected data. Since the replication size was relatively small, each interview was examined in detail and clustered into similar topics. The synthesized topic clusters were further broken down into: major topics that are common among all interviews; unique topics that are still valid but were not repeatable; and outlier topics that were outside the scope of IA (Creswell, 1998).

Each topic was coded as an abbreviation. After all perceived topics were coded as abbreviations, the data was coded by applying the abbreviations to all of the collected data. This established and validated any other new or emerging segments or additional codes that were discovered (Creswell, 1998).

Next emerging topics were consolidated and categorized. Unique descriptions were provided for each category. At this point a final abbreviation was determined and applied one final time to recode the collected data (Creswell, 1998). The collected data was coded for emerging descriptions and themes. The themes were defined by the major topics that emerged from the collected interview data and literature review. Emerging descriptions were also established from the collected data as well and also coded. These emerging descriptions were useful in designing detailed descriptions for case studies (Creswell, 1998).

The fifth step was further advancing the emerged descriptions and themes and how they were represented within the narrative of the case study. This narrative included a detailed discussion on all of the major topics discovered and also evaluated the unique and outlier topics that were also discovered through the coding process (Creswell, 1998).

The final step was the actual interpretation and comparison of the collected data to determine if the emerging themes from the collected interview data matched the emerging themes from the literature review. The details of this interpretation were covered in chapter 5 of this research (Creswell, 1998).

Validity and Reliability

Validity and reliability within qualitative research is somewhat different than validity and reliability within quantitative research. According to Creswell (2009), qualitative validity "means that the researcher checks for accuracy of the findings by employing certain procedures" (p. 190), while qualitative reliability "indicates that the researcher's approach was consistent across different researchers and different projects" (p. 190). According to Gibbs (2007) as cited by Creswell (2009), several steps should be

taken to ensure reliability. They are the following: check transcripts for mistakes made during the transcription process; and make sure there was consistency within the process of coding.

The use of several defined processes and instrumentation helped validate this research, while increasing the reliability and quality of the research. The exploration of possible rivals also helped increase the validity and reliability of this research (Yin, 2009). Using software designed to transcript and code qualitative research and compared against the manual coding techniques utilized by this research further verified this research. The use of multiple evidence sources, creating a case study database and establishing a formal evidence chain improved the construct reliability and overall quality of the research (Yin, 2009). A formal chain of evidence and an established case study database also provides validity for the evidence that was collected and analyzed (Yin, 2009).

Construct Validity

Construct validity was achieved by following a defined process (Yin, 2009). This defined process consisted of using multiple sources of evidence; creating a case study database; and maintaining a chain of evidence. By using multiple evidence sources and data triangulation, this corroborated the proposed research question. Data triangulation addressed potential problems of construct validity when using case studies by using multiple sources to measure the same phenomenon (Yin, 2009).

Internal Validity

According to Yin (2009) internal validity was mainly a concern within explanatory case studies, where event x leads to event y. In the case of descriptive or

exploratory research, a casual situation where a third factor may actually be the influence was not a concern. The bigger risk comes from inferring, which according to Yin (2009) happens in all case studies where an event cannot be directly observed. Using an analytic tactic, such as pattern matching, increases the internal validity of a case study (Yin, 2009).

External Validity

The ability to make generalization within a case study was a threat to external validity (Yin, 2009). To mitigate this threat to external validity, Yin (2009) states multiple replications should be used. This research used multiple cases to minimize any threats to external validity.

Reliability

Yin (2009) states, "The general way of approaching the reliability problem was to make as many steps as operational as possible and to conduct research as if someone were always looking over your shoulder" (p. 45). This research documented all steps taken to complete the research. Tools and repeatable processes helped this research be repeatable and therefore reliable.

Ethical Considerations

Interviews were used to collect data from IA leaders. Their names and names related to their organization were held confidential and anonymous within the results of this research. To ensure all ethical considerations were documented, a request to conduct research communication was sent to each participant. Upon approval to conduct research a second communication for informed consent was sent to participants to sign and record

their consent. All participants were made aware of their obligations and the obligations of this study.

The researcher of this study is qualified as a Ph.D. candidate within the School of Business and Technology at Capella University. The research was supervised by a committee chair / mentor and a committee board consisting of two other board members. The researcher has over 20 years of practitioner experience in IT with the last 15+ years specializing in IA. The researcher has presented at several different security conferences, a Fellow member of the Information Systems Security Association (ISSA), and was a mentor for System Administration, Networking, and Security Institute (SANS).

CHAPTER 4. RESULTS

Proposition One: Computer Science

The first proposition was to examine what IA topics should be taught within CS. The literature review defined six core IA classes that should be included within CS and two other optional classes. The classes defined by the literature review were: cryptography, forensics, network security, ethics, incident handling, and security architecture with risk management and privacy as optional courses (Cate, 2009). The first proposition was to test was if IA practitioners agreed these are core IA concepts that should be included in CS, or if any of these topics should be solely included within an independent IA curriculum.

Cryptography Within Computer Science

The first concept that was reviewed was cryptography. This topic was one of two topics, which one or more case studies believed should be integrated throughout a CS curriculum, but should also be included as an introduction class. The reasoning was cryptography was a critical component in protecting the confidentiality of data, as well as obfuscation, access controls, or not storing or transmitting data within a network (Case 10, 2013). No argument can be made if cryptography should be integrated throughout a CS curriculum or included as a standalone class. Four of the ten cases stated integrated, while three of the cases argued for a standalone class along with one case that felt cryptography should be both an integrated topic and a standalone introduction class (see Appendix B).

Of the cases that stated cryptography should be a standalone class, two out of the three cases (see Appendix B) clearly stated that a cryptography class should be an

introductory class at the beginning of the curriculum, and not an advanced topic that would be covered at the end of the program. Teaching the basics of cryptography was a common theme both in the cases that stated cryptography in a standalone class and cryptography integrated within the entire CS curriculum. Concepts like understanding encryptions algorithms like AES or 3DES, the cases felt were out of scope for required learning objectives within CS.

The theme of analyzing cryptographic algorithms as more of an advanced topic carries over to the two cases (see Appendix B), which believed cryptography was a topic that should be dedicated within an IA curriculum. One of the two cases that felt the scope of cryptography was more around the analysis of cryptographic algorithms. The other case referenced cryptography as sort of a commodity, and that most users of cryptography do not need to know the specific details how it works, rather that it does work.

Forensics Within Computer Science

The second topic that was examined was forensics. Seven out of the ten cases stated computer forensics should be a topic covered within IA, and not a topic within CS (see Figure 13). This was an interesting finding, since most if not all of the published research (Dimkov, Pieters, & Hartel, 2011; Perez et al., 2011; Jensen, Cline, & Guynes, 2006; Cooper et al., 2010; Dark, Ekstrom, & Lunt, 2005; Lester, Narang, & Chen, 2008; Cooper, 2005; Streff & Zhou, 2005; Geoghegan, 2008; Crowley, 2007; Hjelmas & Wolthusen, 2006; Sexton, 2008; Bhagyavati, Naugler, & Frank, 2005; McGuire & Murff, 2006; Vaughn & Dampier, 2007) suggests including computer forensics within CS, SE, or IT curricula. The underlying theme in responses was that computer forensics was a

specialty. All seven cases studied felt an undergraduate in computer science would not have enough background or knowledge or skill set to completely master computer forensics.

Two cases that were reviewed (see Appendix B) agreed that computer forensics should be included as a standalone class within CS. Both cases cited computer forensics was a fundamental concept that all CS majors should have a basic understanding. One of the two cases did agree that mastering computer forensics was a topic better covered in IA, but a basic understanding of forensic components should still be a requirement in CS.

The final case when asked about computer forensics had a different perspective all together:

> I believe that when it comes to computer or digital forensics, that you should teach the offensive side of security as a mandatory component to any sort of digital or computer forensics classes. A lot of organizations focus on finding artifacts and piecing something together like they were a private investigator, but the thing that a lot of people leave out of the view of the private investigator was that. I believe that forensics should be taught as a part of both defensive and offensive security. I believe that forensics, as a standalone without the context of offensive security was not effective. (Case 10, 2014)

The final case agreed that computer forensics as it was taught today was not effective. It not only should be taught as a standalone class within CS, but also integrated throughout CS, not just focusing on defensive aspects, but also teaching offensive aspects as well. This scenario appears to be the only topic where security practitioners do not agree with

academia on what should be taught within computer forensics, when computer forensics should be taught, and how computer forensics should be taught.

Network Security Within Computer Science

The next topic that was analyzed within CS was network security. All cases examined agreed that network security was a fundamental topic that needs to be included within CS. Eight of the ten cases examined felt that network security should be a standalone class within CS. The main theme in these cases was that network security covered various topics and that there were more than enough topics to cover in a single class (see Appendix B).

Two cases (see Appendix B) that were examined felt that network security as a whole could be integrated within CS. One case felt that network security was synonymous with standard operating procedure nowadays. Ten years ago this probably was not the case, but like most if not all of the IA topics discussed today, many would become standard operating procedures and would be integrated into CS.

Ethics Within Computer Science

Ethics within CS was another case where there was no real clear answer (see Appendix B). A majority of the cases examined see a need for ethics within CS, but how the topic was presented was still up for debate. Five of the ten cases examined stated ethics should be integrated into existing CS classes and that a standalone class was not needed. Three of the five cases agree that there was enough content to warrant an ethics class within CS, but expressed concern if a standalone class was the appropriate venue. A common theme from all five cases was ethics in the context of other topics. It was easier

to relate ethical questions in the context of other related security topics. One case did have a slightly different spin on ethics from the other four:

> I don't believe that ethics itself need to be taught as a separate thing, however I believe letting students understand what was the legal boundaries are to what they are doing was important. I believe that the definition that was presented here it tries to encompass that, tries to say that which was legal was ethical. I don't agree with that. So what I would say was understanding your legal boundaries and understanding others perceptions of your actions are was important. And if that becomes the definition of ethics, I think that would be fantastic. (Case 10, 2013)

This case does bring up a valid point; they believe ethics as it is taught today is primarily associated with being legal. Ethics and the way it is being taught needs to evolve, not only to keep up with changes in society, but also to keep the attention of the students.

Three of the cases (see Appendix B) examined agreed with the current methodology of teaching a standalone class within CS. A common theme by these cases was the need for ethics not only within CS, but was a topic that all college majors need to understand. Only two of the cases stated that ethics should not be addressed directly in CS, rather ethics was more of a topic that should be addressed within IA. Ultimately in the end, no one argued that ethics was not important. The argument was more around what specifically should be taught; when ethics should be taught; and where should ethics be taught. Ethics as it was being taught today within CS needs to change.

Incident Handling Within Computer Science

The topic of incident handling within CS was another topic where there was almost a perfect split on practitioners who feel it should be included within CS and

practitioners who feel incident handling belongs in IA (see Appendix B). Six of the studied cases believe incident handling should be a topic taught within IA, while four cases felt incident handling should be taught within CS.

The six cases that felt incident handling was better suited for IA, felt that incident handling was a special skill set. Students would need to know network security, system security, and have some forensics skills in order to grasp the concepts that would be taught in incident handling. A common argument by practitioners supporting incident handling purely within IA and practitioners who support a standalone class in CS was the skill set they feel was required by a student in order master let only understands the concepts of incident handling.

The practitioners that support incident handling within CS are split when it comes to how incident handling should be taught within CS. Two cases (see Appendix B) argue that incident handling should be taught as an integrated topic across various CS classes, while one case argued that a dedicated class was required for incident handling within CS. The final case felt a dedicated introduction class was required, followed by an integration of incident handling concepts throughout the rest of a CS curriculum.

Security Architecture Within Computer Science

There was no clear preference on how security architecture should be taught or if it should be taught within CS (see Appendix B). Four cases showed a preference of not including security architecture as a core-learning objective within CS, rather should be a topic reviewed within IA. A common theme was the concern over if a student would have a sufficient amount of background to truly understand the overall concepts of security architecture. For example Case 9 states:

> You can't be a security architect until you've been doing a role that was engineering operational understanding ... into the point of being enrolled where you're doing break-fix and not really understand how a network and the systems work together in some level of business intelligence, how businesses were built middleware, and the whole enterprise ... you have to have an enterprise architecture of that too. (2013)

This was a valid point, and was a low level overall theme in many of the cases. Information assurance was a compilation of various studies. It was difficult to teach a student the concepts of security if they do not have a thorough understanding of rudimentary concepts of networking and system administration.

Three of the cases (see Appendix B) examined felt security architecture should be a standalone class within CS, leaving two cases preferring that security architecture should be integrated throughout the CS curriculum. One case felt it should be a standalone introduction course, and then integrated throughout the rest of the CS curriculum. The single case of both standalone and integration was somewhat of an outlier. The cases that selected a standalone class for security architecture within CS all agreed a standalone class near the end of the program was better suited than an introduction class. A similar concern was raised on a standalone class of security architecture compared with the cases that felt security architecture was a topic better covered in an IA curriculum. Security architecture as a whole comes down to experience. A majority of the cases agree that security architecture should be an upper level class or moved to a dedicated IA program to be the most effective.

Risk Management Within Computer Science

The collected data does support the proposal of risk management as an optional class within CS. Seven out of ten cases felt risk management was concept that was better covered in a dedicated IA program; while only two cases felt risk management could be integrated into existing classes within the CS curriculum (see Appendix B). A single case argued that risk management should not only be taught as a standalone class within CS, but also integrated throughout the CS curriculum.

The common argument in most of the cases was risk management was a specialized skill set, much like a security architect. The overall knowledge required by a risk analyst was rather broad. A thorough knowledge of all the concepts covered in CS was required in order to properly understand, let alone manage risk. According to Case 9, "It's also a gap because people have less interest in that area of subject-matter expertise. You find a lot of people just want to be technical and that was in the process piece" (2013). Case 1 further supports the want for more technical classes within CS, who states "[Risk management] was a complex enough of a topic even people who have been practicing it for a long time don't see it beyond the technology" (2013). CS students are drawn more to technical classes, but there was a need to understand risk management. Many of the cases examined support the need for risk management, but see risk management as an optional class for CS students or a topic within IA rather than a requirement, because not all students will show an interest in risk management, nor do they have an interest in becoming an expert in risk management.

Privacy Within Computer Science

Privacy within CS was another topic that varies greatly in the responses that were collected (see Appendix B). Half of the cases examined believed privacy was a topic that was better taught within IA. Two cases (see Appendix B) felt privacy was a standalone class within CS. Two other cases (see Appendix B) felt privacy should be integrated throughout the CS curriculum, while the last case argued privacy should be included as an introductory class in CS, but also integrated throughout the CS curriculum. One common theme among all the cases was the scope of privacy. Many believed privacy and ethics should be combined into a common introductory class.

Proposition Two: Software Engineering

The literature review defines several IA concepts that should be addressed throughout the SE curriculum. They are: buffer overflows; SQL injection attacks; cross-site scripting; cross-site request forgery; click jacking of code; broken session authentication and session management; insecure direct object references; security mis-configurations; insecure cryptographic storage; failure to restrict URL access; insufficient transport layer protection; and invalidated redirects and forwards. The literature review also defines two core IA concepts that should be included in SE. They are: cryptography and software risk and project management. Proposition two was to test if IA practitioners agree with incorporating common software vulnerabilities throughout a SE curriculum and appending two core IA classes within the SE curriculum.

Cryptography Within Software Engineering

Out of the ten reviewed cases, nine cases agreed cryptography should be included as a core topic within SE (see Appendix B). The disagreement comes when asked how to

include cryptography within SE. Five of the nine cases believed cryptography should be integrated throughout the SE curriculum. When each of the five cases was asked to elaborate, all of them gave the same answer. Cryptography was a core-concept that needs to be covered. Each case did clarify that cryptography methods could be integrated. For example: when should cryptography be used; what information needs to be encrypted; where should cryptography be used; and why should cryptography be used.

The four cases that felt cryptography should be a dedicated class within SE, scoped cryptography as how does cryptography work, such as cryptographic analysis. This leads to the final case, which believes cryptography should be taught within IA. The study of cryptographic algorithms goes beyond the scope of a traditional SE skill set. According to Case 8, "I think cryptography was far beyond the scope of what I would expect of any software engineer. It's the subject matter expertise that I could see the security guys educating the app guys constantly in the professional workforce versus in an academic space" (2013).

Common Vulnerabilities Within Software Engineering

As with cryptography, almost all the cases examined agreed common vulnerabilities needs to be addressed within the SE curriculum (see Appendix B). Eight of the cases examined believe common vulnerabilities should be integrated into every class within the SE curriculum. Whether discussing input SQL injection directly, or covering a common vulnerability like input validation in a lab, there are ample scenarios in which common vulnerabilities can be brought up and repeated over and over again.

One case (see Appendix B) argued that a standalone class should be established at the beginning of the SE curriculum. They argued by solely integrating, the emphasis and

importance of understanding common vulnerabilities would be lost. They did agree that ultimately integrating common vulnerabilities into all SE classes was also an important aspect as well.

One outlier was a single case (see Appendix B) that argued common vulnerabilities was more of an IA concept. They argued bad coding practices are more of a corporate culture issue than an academic issue. Students may learn about common vulnerabilities within their education, but ultimately the code they produce was influenced by the corporate culture and business vertical the student ultimately ends up at. Teaching students about common vulnerabilities may have some impact, but ultimately educating an organization will have more impact on society as a whole.

Software Risk and Project Management Within Software Engineering

As with the other two scenarios examined within SE, a majority of the cases examined believe software risk and project management should be covered within the SE curriculum (see Appendix B). The argument was how software risk and project management should be taught. Half of the cases evaluated believe software risk and project management should be integrated, while the other half of the cases evaluated believes software risk and project management should be a dedicated class within the SE curriculum.

The common argument for integrating software risk and project management was software risk and project management was a fundamental concept. As Case 10 states, "Let me rephrase the question. Should you teach software coders how to manage their time?" (2013). When put in that perspective, the concept of software risk and project management does seem like common sense.

The common argument for having a dedicated class for software risk and project management was that all of the different software development philosophies should be covered at once, and not spread out across several classes. For example, Case 7 states:

> To do that now you probably should have feature a couple of different project management sort of philosophies you know where it's the formal inbox stuff or whether it gets into a little bit more agile development and some of these other things which you deal with in a little different manner and handle in different ways. You can see different concepts of how you might be asked to work on a team on going after the fact, that I think would be important. (2013)

The final two cases (see Appendix B) take the argument more from a perspective of software risk and not project management. Both cases feel software risk was a specialty to IA. Both feel you need to have software risk specialists that see software risks from every different aspect within an organization, such as development and infrastructure. Teaching a software engineer this skill set would be ineffective, since they typically only see the software development side of the argument. It was the IA specialist in software risk that was trained to see the complete picture and was the person who was ultimately responsible for the security of the system.

Proposition Three: Information Technology

The literature review defines several core IA concepts that are combined to create a set of classes specific to IA within IT. These core IA concepts are: fundamental aspects; security mechanisms and countermeasures; operational issues; policy; attacks; security domains; forensics; information states; security services; threat analysis models; and vulnerabilities. Proposition three was to see if IA practitioners agree that these core IA

concepts should be included within IT, or if any of these concepts should be taught in a dedicated IA curriculum.

Fundamental Aspects of Security Within Information Technology

All ten cases that were reviewed agreed that fundamental aspects of security should be included within the IT curriculum (see Appendix B). The methodology that should be used was perfectly split in half. Five of the cases examined believed fundamentals aspects of security should be integrated throughout the IT curriculum, while the other five cases examined believed fundamentals aspects of security should be included as a standalone class within the IT curriculum.

A common theme within integration of fundamental aspects of security was the concepts are too broad. Each case was given the definition as defined by Machonachy, Schou, Ragsdale, and Welch (2001) who state, the fundamental aspects of security includes: Security Services (Availability, Integrity, Authentication, Confidentiality and Non-repudiation), Security Countermeasures (Technology, Policy, and People), and Information states (Transmission, Storage, and Processing). Each of these topics was asked as a separate question to each of the cases, in order to validate if a specific topic should be called out within IT.

Five of the cases believe that fundamental aspects of security should be a standalone-dedicated class within the IT curriculum. The general consensus from the cases was that fundamental aspects of security are a core and rudimentary part of IT. All students graduating from a college or university with an IT degree should thoroughly understand fundamental security concepts and how that impacts or will impact their

career in IT. Three of the five cases specifically stated that fundamentals aspects of security should be a mandatory introductory course within IT.

Security Mechanisms and Countermeasures Within Information Technology

Security mechanisms and countermeasures were not as definitive as fundamental aspects of security (see Appendix B). Seven cases believe security mechanisms and countermeasures was an important concept that should be addressed within the IT curriculum. Three of the cases (see Appendix B) believe security mechanisms and countermeasures should be covered in an IA curriculum and not within IT. Unfortunately no theme can be established from these three cases. When each case was asked to elaborate, each case stated it was a gut feeling and could not further describe why they had a gut feeling.

Four of the cases that felt security mechanisms and countermeasures should be included within the IT curriculum as a standalone course. The concern with the cases was scope of the definition. Each of the cases was given the definition of security mechanisms and countermeasures from the ACM (2005) which states: security mechanism and countermeasures includes the following: cryptography, authentication, redundancy, and intrusion detection. The specific concern was each of the sub-topics could be addressed differently, depending on the scenario. A dedicated class could address each of the different scenarios (viewpoints).

Two of the cases (see Appendix B) examined believed that security mechanisms and countermeasures should be integrated within the IT curriculum, but no underlying viewpoint could be determined from the two cases. One final case (see Appendix B) believed that security mechanisms and countermeasures should be both a dedicated class

and integrated throughout the IT curriculum; arguing that security mechanisms and countermeasures was the highest risk today in most organizations, and should be the direction that all organizations should adopt. The concept of security mechanisms and countermeasures was an important topic that requires a dedicated introductory class, followed by an integration of the concepts in the different scenarios that are address across the different coursework presented within an IT curriculum.

Security Operations Within Information Technology

The concept of security operations within IT was somewhat interesting (see Appendix B). Three of the ten cases reviewed believed it was a topic that should be addressed within an IA program. The common feeling was security operations are something that was more honed on the job. For example Case 7 (2013) states:

> And my rationale around that was that you can't really develop the context of an organization in the security component, the operational parts of it are...I don't even know how you would address the theoretical operational issues around that.

The other seven cases argue that operational security should be included within the IT curriculum. The problem was how it should be included. Five of the seven cases believe operational security should be integrated throughout an IT curriculum. The common argument was security operations by itself do not have enough content to warrant a standalone class. Security operations are a broad discussion on operational aspects of security, which touches various subsets of security. Integrating it within other classes would not only enhance those classes, but also put security operations into a context that can be easily remembered.

The last two cases (see Appendix B) believed security operations should be a standalone class. No definitive common argument can be made. One case felt it was an introductory course, which could be combined with various other security concepts, while the other case related security operations back to a capture the flag scenario. Case 9 (2013) states, "You teach them the operational concepts and how things work and then you have red team, blue team exercises to actually show them how, within a technical environment, that rolls out." This was an interesting viewpoint, and one of a few comments that directly specified Capture the Flag (CTF) exercises.

Policy Creation and Management Within Information Technology

Eight of the ten cases (see Appendix B) agreed that Policy Creation and Management should be included within the IT curriculum. Six of the cases believed that Policy Creation and Management should be a standalone class within the IT curriculum, while the other two cases felt is should be integrated throughout the IT curriculum. Only two cases believed that Policy Creation and Management should be taught solely within an IA program. Unfortunately the two cases (Case 2 and Case 9, 2013) were indecisive. When asked to elaborate on their decision, both cases agreed it could be included as an optional track but not a required course for an IT curriculum.

Attacks Within Information Technology

The concept of attacks within IT was mixed (see Appendix B). All cases that were examined agreed that attacks, as a core topic should be included within IT. Of the ten cases that were evaluated, six of them believed the topic of attacks should be integrated throughout an IT program. A common theme that was presented was the need to understand attacks within the various aspects of IT. For example, several of the cases

specifically called out network security. A network administrator needs to understand the different attack vectors so he or she knows how to defend against them. A sub theme was the ability to understand attacks by applying them or seeing how they apply to the different aspects of IT. The six cases evaluated felt having a dedicated class on attacks would weaken the understanding, since a dedicated class on attacks would concentrate more on the attack and not on the scenario. The cases that were evaluated consider putting attacks in context of the scenario crucial.

Three of the cases felt attacks should be a dedicated class within IT. The general consensus was attacks should be discussed as an independent topic, but perhaps as part of a set of IA classes within IT. Two of the cases referenced attacks as a topic within an operational security class, which relates back to teaching attacks within the context of the scenario; in this case operational security. One case (see Appendix B) examined believed the concepts of attacks should not only be a dedicated class within IT, but also integrated throughout the IT program. According to Case 10 (2013):

> Teach the offense. The more that you teach the offense, the more people are going to think about the defensive measures. While many people will get scared at the idea of teaching people how to do malicious things or things that could be considered malicious, I believe there was more benefit to people, more people thinking about how to prevent these types of things because they know how to do them. Integrate that, heck integrate and have a standalone introduction.

According to Case 10 (2013), teaching attacks was teaching someone the offensive mechanisms required to be a security practitioner today. The same case feels too much

emphasis was being done on the defensive mechanisms and not enough on the offensive mechanisms.

Security Domains Within Information Technology

The concept of security domains has different meanings and connotations depending on the person being interviewed. To standardize on the definition, each case was given the same definition by the ACM (2005) which defines security domains as the following aspects: human-computer interaction, information management, integrative programming, networking, program fundamentals, platform technologies, system administration, system integration / architecture, social issues, web systems, and physical plant (see Appendix C).

Only one case (see Appendix B) felt security domains were a dedicated topic within IA. When asked to elaborate, they based it more on a gut decision and personal preference. The other nine cases were split on whether to integrate security domains throughout an IT program or establishing security domains as a dedicated course within IT. Four of the cases believed security domains should be integrated throughout an IT program. The underlying theme, all cases felt the concept of security domains were topics that were generally already taught within IT. For example, according to Case 7 (2013), "It feels to me like there was a lot of overlap with what should be expected in an IT, I'm an IT professional so I would, with an IT degree." Case 10 (2013) also supports the integration of security domains within IT:

> This should be an ongoing discussion throughout...this should be an underlying theme throughout the program. Getting people to think in these domains was similar to trying to get people to think for the CISSP and the common body of

knowledge. Most people don't. I would integrate these into not just each class, but establishing this terminology and establishing this way of looking at the different domains was important to people new within the industry as well as people who have been in the industry a very long time. When I am sitting here looking at this definition, I'm like the call out web systems specifically. They call out social issues specifically. While I think of those things, I would not think of those as security domains. I am intrigued as well as a little sad I didn't think of those ahead of time.

The other five cases that were examined believed security domains should be a dedicated class with IT. The general theme was that security domains should be part of a fundamental course on IA concepts within IT. When each case was asked to elaborate, they all agreed it was an important topic, but the concept of security domains could easily be combined with other IA topics within a fundamental class.

Forensics Within Information Technology

Similar to the perceptions captured in CS, seven of the ten cases (see Appendix B) felt Forensics belonged solely in an IA program, and should not be included within the IT curriculum. Of the other cases, only one case (Case 10, 2013) believed Forensics should be included throughout an IT curriculum, but even then Case 10 (2013) had some doubts:

> Again I would take this back to a mix of our previous law discussion and offensive security. I understand there are the three perspectives of this interview. I believe it does not belong specifically within information assurance. I believe this is very cross-domain - very cross practice. You have to know the offense to know the defense. Forensics is when defense went wrong. If you don't have the

> knowledge base behind that to understand what the offense is, you are going to be ineffective in forensics. When you look at it from the law enforcement perspective or the military perspective, why do we do drills? We do drills so we muscle memory when something bad happens and we need to respond. When I do intrusion analysis, the process that I learned at MITRE way back in the day...still comes to mind. It's to understand everything. Understand all of your different areas. Then I revert to my PI background, and then I revert to all of this other stuff. Bring those all together. If I did not have that background, I would not be effective in forensics. You are kind of in the position of how do you shove that in people's heads in 3 years?

Teaching an undergraduate degree in IT is more than just IT classes, Liberal Arts classes need to be taken to round out a student's education. Trying to teach a specialty IA topic on top of required IT courses could be a lot to remember and comprehend. The problem here is timing. Students need to have a thorough understanding of computers and systems, which is typically taught throughout an IT program. Trying to teach Forensics at the beginning of an IT program, the concepts would be lost. Forensics would need to be taught at the end of the IT program, but even then there may not be enough time.

Information States Within Information Technology

All cases that were reviewed agreed information states should be included within IT (see Appendix B). A majority of the cases reviewed believed information states should be integrated throughout an IT curriculum. The general theme was similar to security domains. The concept of information states was a fundamental aspect of IT, though not specially called out. Different IT classes focus on the transmission, storage, or processing

of data, but typically the states are not referenced, rather the technology such as network infrastructure. According to Case 7 (2013), “I think that there's a gap in how people are coming out with an IT degree, that they don't understand the fundamental pieces of data and where data exists in the world”. Seven out of the ten cases agree that information states was a core concept that needs to be addressed throughout an IT curriculum.

The other three cases examined felt security states should be a dedicated class within IT. No real common theme can be developed on why. One case felt it should be integrated within IT, but then felt because of cloud resources there was enough concepts to cover a dedicated class was then warranted. Another case referenced state as a transactional state.

Security Services Within Information Technology

Security services are another concept that more cases agree should be included within IT (see Appendix B). Only three cases felt security services should be a topic covered within IA. One case summarizes the feeling of all three cases that agree security services were an IA topic:

> So program, policy, risk management, architecture, certification and accreditation, that stuff was core stuff within information assurance. Probably more peripheral to a general IT tracks. But when you get into firewalls, Intrusion detection, certificates, that sort of stuff; maybe that was a little more integrated into at least a network track that was more IT general.

The three cases (see Appendix B), which felt security services belonged in IA, were not completely convinced security services solely belonged within IA, and indirectly support the integration of security services throughout IT.

Six of the ten cases evaluated believed security services should be taught as a dedicated course within IT. A common theme among all cases was security services are a core concept and should be part of a fundamentals course. Several of the cases referenced security domains and security services as a fundamental introductory course within IT. Only one case (see Appendix B) that was examined felt security services should be integrated within IT.

Threat Analysis Models Within Information Technology

There was no clear answer where threat analysis should reside (see Appendix B). Four of the ten cases examined felt threat analysis was a topic within IA. The main belief was threat analysis was a specialized topic and it should not be expected by an IT graduate to be able to analyze threats.

Another four cases felt threat analysis should be an integrated topic within an IT curriculum. Case 10 (2013) best summarizes why threat analysis should be integrated:

> Integrated. 100 percent. Knowing what your threats are gives you the knowledge and the scope to understand what controls you need to put in place. If people don't know what they are defending against, I think that they cannot be effective defenders. Everything that we do...if people had the awareness of how their (going back to software engineering) if coders knew what they were defending against; talking about the state sponsored attacks; the insider threats; advanced persistent threats; whatever; I think they would code differently. I think that

> system administrators and those working on the information technology deployment side would think differently. The moment you explain to people that just like...ooh who was it...Sam sheep dog and Wile E Coyote, the attackers out there go to work every day just like you and me do. They get to work and they clock in, and we are targeted. It's nothing personal. It was just business. I think that those threats that people don't think about are the ones that are causing us to fail today. We gave gaping holes in our critical infrastructure. We have within our organizations, we have so many holes that we can't even begin to enumerate them. We have systems that have been around for 30 years; you think they have been patched? No. You look at the core infrastructure of the Internet, how many of those routers have 1000 days of uptime? Today 1000 days of uptime was not cool. It's no longer that my system has been up for 1000 days; it was I have not rebooted or patched that thing in over 1000 days. Three years of not patching was not cool. That's just not going to work. You have to know what your threats are in order to know what you are defending against. Furthermore, without that knowledge you cannot be effective.

The final two cases (see Appendix B) examined believed threat analysis should be a standalone class within IT. Specifically threat analysis should be a part of a fundamental IA class within IT, which includes other topics like security domains and security services.

Vulnerabilities Within Information Technology

The last topic reviewed in IT was vulnerabilities. Only two cases felt vulnerabilities should be covered within IA (see Appendix B). No common arguments

can be made why these cases felt vulnerabilities were an IA topic. When asked to elaborate, both cases said it was more of a gut feeling.

Six of the ten cases felt vulnerabilities should be covered as an integrated topic throughout an IT curriculum. The common theme that can be applied was the complexity of vulnerabilities and discussing them without the context of other IT topics. Many felt that talking about vulnerabilities as they relate to an IT topic would be easier than having a dedicated class. For example Case 10 (2013) states:

> That was integrated. That needs to be integrated. I would partner that up on the software engineering side and back to the discussion of the buffer overflows, those core things. Those two things should be intertwined. If they are not closely intertwined, then that means we are still treating programmers, developers, engineers as standalones. Not as people that are integrated as part of the larger security organization.

Another common sub-theme was the scope of vulnerabilities. Three of the cases made comments to the effect that vulnerabilities are important to discuss, but they felt a whole class within IT would be overkill. The last two cases (see Appendix B) examined felt a standalone class within IT was required, but vulnerabilities should be part of an IA fundamental class that covers other topics, such as security services and security domains.

General Case Overview

At the end of each interview, each case was given the opportunity to discuss any topic they felt was not covered in the static questions that they were asked. From this, five recurring themes were established.

General Knowledge

Several of the cases (Case 1, Case 3, Case 4, Case 8, Case 9, & Case 10, 2013) all expressed concern over the general knowledge an information assurance major could obtain. None of the cases evaluated graduated with a degree in CS, SE, or IT. Each case's experience came from on the job learning. Each case came from a different background, such as network administrator to mainframe operator. Each case started in a non-security field, learning the ins and outs of each job function.

The concern was IA students today are and will be lacking the general knowledge acquired through on the job learning. All agree a two-year associates degree does not come close to covering everything that was required for an IA professional today. Several of the cases even felt an undergraduate degree in IA was not a sufficient amount of time for a student to truly master the concepts within IA, and also obtain the background knowledge required to know the systems and infrastructures they are trying to protect. A few of the cases felt that IA should be a graduate program only, and that all students need to have either a CS, SE, or IT degree to get the sufficient background required for the systems and infrastructures that are going to be protected. A question was posed by one of the cases: how can a student understand forensics, if they do not understand the underlying operating system or file system.

By no means are the cases stating IA should not be included within CS, SE, or IT, or that they should have masterly level competence in IA concepts (see Appendix B). The primary focus should be to understand the concepts within CS, SE, or IT but applying concepts within IA throughout a CS, SE, or IT program that can be easily remembered. There was still a need to have experts in IA that specialize in forensics or incident

handling, but those skill sets are very specialized and are something that could be taught in an IA program. An information assurance professional today needs to be a jack-of-all-trades, but they need to have a thorough understanding of the underlying systems and infrastructures they are protecting.

Cross Discipline

A second underlying theme within several cases (Case 1, Case 4, Case 6, Case 7, & Case 10, 2013) was the need to have cross discipline integration. This cross discipline integration was defined in two different approaches: 1) Require a set of business classes within the CS, SE, or IT program; or 2) Require other programs to require a basic IA class as a required class.

Building a relationship with the business was critical, and understanding how to build that relationship was extremely important. For example, Case 1 (2013) states, "I've spent time in the infrastructure and networking and have had security as a part of those, so when I am a security professional I can be of more service not only to the rest of the was team, but as a partner in delivering a service to help a business. Technology was great, but without the business it's no good." Case 6 (2013) further supports integrations with business concepts, stating:

> I would almost like to see information security coupled with a business type degree. I think there needs to be that combination of something other than just information assurance or information security. I think there has to be other things that factor into it because today's security professional should be a business person, should be a person that knows how to work with the business, understands

how businesses work, understands the budgeting process and business goals and business needs.

Case 7 (2013) further elaborates on cross discipline integration:

> I think some of the other fundamental IT and computer science components are so fundamentally underlined what's there, having a good solid base in that at it from a business or you know a human science maybe area, that understands the human behavior side of it, and inclined to have solid risk management and other things, requires law degrees and things like that, that might be able to come back in and from the management program.

Finally Case 10 (2013) sees cross discipline integration from a different perspective:

> To your question, it must be integrated. I don't believe it was a standalone profession. I believe that every degree program should include this. I'm not talking purely the techie ones. If you go for your MBA, you dam well better know about security. You better know what you as a businessperson are going to be communicating to handle the data that you are working with on a day-to-day basis. You need to know the legal aspects of it. You go to be a lawyer, you dam well better know how to handle your clients data. You should know that if you lose the USB drive that has your entire defense on it, you are probably screwed. Yes we have the concept of discovery, but that doesn't mean that you should be sharing your private client discussions.

Critical Thinking

The next compiled theme was critical thinking skills. Three of the ten cases examined believed CS, SE, and IT graduates today are lacking critical thinking skills

(Case 2, Case 4, & Case 7, 2013). The question was this specific to IA or a general issue. Case 4 (2013) had this viewpoint:

> Critical thinking skills, how do you gain that through four years or six years, even? I think that's a syndrome of just where the discipline is; it's very young. Twenty-five years ago you would have said, maybe longer than that ago, you would have said that about computer programmers. They learned on the job until the 60's or early 70's there weren't really computer science programs. So to me part of that was just where the profession was as a discipline, it's very young. In 20 years it'll be very different.

Case 7 (2013) further elaborates on critical thinking:

> Leadership in my mind includes those critical thinking skills. Because too many...and it's interesting beyond just information security not just with the IT people. In the hospital we have a variety of people that are in IT that came from different backgrounds. And some are came from a nursing background and then moved away into IT through some techno courses; just had sort of moved into and sort of figured it out and were allowed to that freedom. But I think that they are sort of like in their own little world, and they just can't see outside of what they're sort of doing and that's just the ability to understand where you are in the ecosystem was so important because if you're doing it one way, and if based within an application that's fine, but if it was its own thing, then it's going stand alone, then it's totally different.

Critical thinking skills are a vital skill set that all college graduates require, whether in CS, SE, IT or IA. Today's complex technical ecosystems require people who can think through issues and develop solutions to those issues.

Apprentice and Mentorship

Another common theme between several of the cases was the discussion around mentoring and apprenticeship. Only one case made an argument for IA education prior to college. Case 5 (2013) states:

> I would have to say that there most certainly should be both an undergraduate degree and a graduate degree and even maybe introduction of information assurance at the high school level. I think computing is so ubiquitous at this point. We're seeing devices being used by children on a daily basis, integrated into the school systems. They don't really have a clue. There should be integration of aspects of information assurance, information education from the middle school level on up but most certainly an undergraduate degree would be very appropriate as well as a graduate degree, most certainly, and I say that from a practitioner perspective in that we have such a shortage of information security professionals right now at all levels, whether it's highly technical to reading the architecture, and it's not just this community, it's in other communities I've been in and had the same discussions with whether Des Moines, Chicago, Brussels, London, Dubuque, it doesn't matter … Austin.

This falls out of scope for this paper, but Case 5 (2013) does have a valid point and the starting argument for perhaps a new learning model. Case 9 (2013) continues the argument for a new approach for learning through apprenticeship. Case 9 (2013) states:

> I think it's all about how you transition someone from the academic environment into the corporate environment and you have to work with Fortune 1000 or Fortune 2000 companies to develop a program that allows someone who wants to be an information security to do what's necessary to have that knowledge be effective in their role, in a nutshell. That's kind of a brainstorm more than anything because I never really thought about it that way, but I think the intern program at [anonymous] was extremely effective in training those guys and leading them out to ... I mean, if they were only ever going to be a project manager after they're done with the intern program, you knew they were only going to be a project manager. My intern, he picked up that firewall stuff on a technical level very quickly and I knew he had a really great technical skill set just after walking him through that, to hit the ability to kind of figure that out. That 2 years helps him weed out ... I shouldn't say weed out, it helps you decide what that person is suited for within the IT environment, and if they're suited for security, cherry pick them out of that intern program and bring them in security.

This supports a more traditional intern program, but Case 10 (2013) pushes the concept of an intern as more of an apprentice model. Case 10 (2013) elaborates:

> The educational model we have today cannot keep up with technology. I do not believe that it is possible for the level of complexity that exists in the Internet today, to be captured within a short period of time. When I talk about a short period of time I am talking about a single class. I am not talking about trimester or semester, quarters, whatever. One class cannot do that. I also believe that from a program standpoint, it is challenging to give the lessons learned a neither proper

> introduction nor mastery. What we are really looking at today and where I would take information assurance is a proper apprenticeship program. I am talking about a 3-year program as an apprentice, working with or for someone like me, or the John Strands of the world or the other experts we have in the community. Your apprenticeship leads to your journeyman; your journeyman leads to your masters; your masters lead to your proper expert.

IA concepts are constantly changing. Keeping students current on these concepts is one concern. Keeping instructors current on IA concepts is also an issue. Case 3 (2013) supports this concern:

> It's hard to teach security and understand it without having hands-on knowledge… You know someone whom probably better than I do but I think that's what we need is a true security professional teaching a class not from an academia from but from hands on experience. If you get some instructor who teaches pure from academia, the program will fail the student because the student will not understand the real world. They'll always understand, "Well, this is the book says." You can't teach from just books. That's what I feel.

Between concerns of covering too much information, developing critical thinking skills, and a constantly changing environment, a new argument could be made that a traditional undergraduate degree in IA may not be an ideal solution.

Historical Perspective

The final theme discovered from all interviewed cases is the need to teach a historical perspective. Case 5 (2013) expands on the need for a historical perspective:

> … On one condition, and that condition is that the history of security or information assurance is taught as well. If you're a mathematician you can go down a very specific path and have no knowledge of the rest of the world or how it operates, but you will understand its context if you understand the history of mathematics, because it fits within the world and what's going on in the world. Without that understanding of that historical perspective, the person who comes out with that degree won't really have … it will take years of experience to really understand how they really fit in the world, but if they have an understanding of the world and how this fits into the world, they'll come out of it with understanding of how they fit in the world. I don't know if I articulated that well but from the mathematical perspective, and once you know who Evariste Galois was, and how he died, and why he died, and what period of time he died, and what was going on at the time when he died, a very dramatic story of a young man in the 19th century who was killed in a duel at 19 and then night before he wrote his most important paper on advanced algebra that had affect for hundreds of years. Without an understanding of the history of mathematics we wouldn't even know who this person was. We're just presented with a textbook. The chapter is nicely refined and defined which seems completely, I would just say to some degree, falling within a framework and yet there's real life that went into that evolution of that notion so if history isn't taught along with this, it's of no value.

The argument for teaching a historical perspective is further supported by Case 10 (2013), who states "There is always something going on in this industry. And it may be

stuff we talked about 20 years ago. It probably is, but we are still going to come back to it. As in the series Battlestar Galatica, this has all happened before and it will all happen again. Information assurance is an interesting discipline." No matter what discipline, and argument can be made that a historical perspective needs to be included.

CHAPTER 5. DISCUSSION, IMPLICATIONS, RECOMMENDATIONS

Discussion

An unexpected discovery from this research was the level and type of education disclosed in each case. None of the cases examined had completed a degree in CS, SE, or IT. This is somewhat to be expected, considering the maturity of IA as a field of study. Most, if not all of the cases examined, learned IA concepts on the job. What was not expected was the lack of CS, SE, or IT degrees obtained. Since this research was to address core IA competencies within CS, SE, or IT curricula, a thorough or in-depth knowledge within CS, SE, or IT was not needed. This may have had some influence on whether an IA concept should be integrated throughout a curriculum or a standalone class, due to the lack of knowledge of existing courses. This concern is somewhat mitigated since at least half of the cases examined were also adjunct faculty or had taught at one point at a local college within a CS or IT program. Specific details unfortunately were not disclosed, since this information was captured in an open-ended question at the end of the interview and not requested specifically.

The research question this research addressed: What IA core competencies should be included in CS, SE, and IT curricula? This was broken down into three propositions.

Proposition One: Computer Science

The literature review defined six core IA classes that should be included within CS and two other optional classes. The classes defined by the literature review are: cryptography, forensics, network security, ethics, incident handling, and security architecture; with risk management and privacy as optional courses (Cate, 2009). The first proposition was to test if IA practitioners agree that these are core IA concepts that

should be included in CS, or if they should be solely included within an independent IA curriculum.

Cryptography

Out of the ten cases explored, eight of them agree that cryptography is a core IA topic that should be taught within CS. The question that remains is how it should be taught. Exactly half of the cases felt it should be integrated, while the other half felt it should be a dedicated class. The proposed solution is to include an introductory class of security concepts within CS, with cryptography covered as a core concept. This is supported by several of the examined cases (Case 4, Case 5, Case 8, Case 9, & Case 10, 2013). Then integrate cryptography throughout CS within cases, examples and assignments. Simple attacks like SSLStrip (Marlinspike, 2009) are common day practices, which push the need to default to secure communication across all infrastructures.

The agreement for a need of cryptography in CS by a majority of the cases examined is not a surprise. In a personal observation, when the researcher recently interviewed candidates for a security analyst position, he noticed a lack of understanding of basic cryptography concepts, such as Secure Socket Layer (SSL). Each candidate interviewed could not explain in detail how SSL functioned. Most IA leaders know the importance of message level encryption and transport level encryption, so requesting new college graduates to know cryptography is not a surprise.

Forensics

Seven out of the ten cases examined felt forensics was too specialized of a skill set to be included within CS. The largest concern brought up when discussing Forensics

was the lack of thorough knowledge of systems and infrastructures. Without having a good understanding of the systems that are being investigated, it is hard for students to understand all the ins and outs related to the hardware and software that is being examined. This is supported by several of the cases (Case 3, Case 8, Case 9, & Case 10, 2013) that are also adjunct faculty and teach security classes within CS, SE, and IT programs.

From the researcher's personal observation, this was the expected result. The researcher teaches various undergraduate classes, of which a set of computer forensics classes is included. The researcher noticed that many students did not have the required skill sets to complete most, if not all of the required labs. Simple skills like command line parameters were not known or not understood. Labs over time evolved and became simpler, either through pre-scripting various commands or documenting step-by-step instructions. This unfortunately can cause students to miss critical thinking skills and rely on canned scripts or perfect textbook scenarios and shows students are not ready to fully understand any advanced Forensics tasks. The proposal is to drop the concept of Forensics from the CS curriculum, in favor of including it as a core competency within IA.

Network Security

All cases examined agreed Network Security should be a core component within CS. Eight of the ten cases agreed that Network Security should be a standalone class within CS (Case 1, Case 2, Case 4, Case 5, Case 7, Case 8, Case 9, & Case 10, 2013). The other two cases examined compared it to cryptography. They agreed a dedicated class could be done, but both cases felt network security is a fundamental concept that

needed to be repeated throughout the CS program. The proposal is to establish a standalone class within the CS curriculum that focuses strictly on network security.

This finding is exactly what the researcher expected to see. Many CS programs today do include a network class, but most of the time this class is not specific to network security. Many technical practitioners associate security to the installation and configuration of firewalls and intrusion detection systems. It is a natural progression to migrate current networking classes within CS to a secure networking class.

Ethics

Eight of the ten cases examined felt ethics should be included within CS. Five of the eight cases felt Ethics should be integrated throughout the CS program, while the other three cases believed a dedicated Ethics class was more appropriate. The consensus with Ethics as a standalone class is that the topic is typically covered as a mundane concept and is quickly forgotten because students do not have any interest in learning about Ethics. This can be compared to teaching Math or English within every Liberal Arts requirement. The proposal is to spread the concepts of Ethics throughout the CS curriculum. The thought is a student will be exposed to Ethics several times and will hopefully retain the values. By also incorporating Ethics throughout the CS curricula, Ethics can be taught within the context of CS, and perhaps spark debate and discussion, rather than being a standalone mundane topic that is seen as a requirement.

Incident Handling

Six of the ten cases examined felt Incident Handling should be left out of the CS curriculum and taught within a dedicated IA program. Similar to Forensics, the cases that were examined agreed that Incident Handling was a specialized skill set within IA. This

finding was a bit of a surprise for the researcher. Many IA practitioners would agree that it is not a matter of if an organization will be compromised; it is a matter of when they will be compromised and how quickly an organization can respond. Having an educated work staff educated on incident handling should be a requirement. The only explanation for including incident handling as an introductory course would be that not everyone would be an incident handler. Typically an incident handler is a dedicated resource with specialized training. The rest of the incident response team only needs to be familiar with the process, thus the perception that incident handling only needs to be a concept covered in an introductory IA course. The proposal is to have Incident Handling be taught within CS as an introductory level class with other IA concepts and covered at a high level.

Security Architecture

Security Architecture within CS is only slightly preferred as a core topic within CS, with only six of the examined cases stated it should be included as a standalone class or integrated throughout a CS program. This is another topic that is better suited within an introductory class along with Incident Handling. Overall there were really no strong arguments for or against including Security Architecture within CS. The result is what was to be expected. Having a dedicated class specific to Security Architecture would impact what other classes could be offered in a CS curriculum, specifically core CS concepts which should be covered in greater detail. CS majors only need to have a working knowledge of Security Architecture; they do no need to be experts. Covering Security Architecture in an introductory class would give CS majors the needed knowledge required, while minimizing the direct impact to other core CS concepts. The

proposal would be to include Security Architecture as a concept within an introductory class at the beginning of the CS curriculum.

Risk Management

One surprise discovery in this research was the response related to Risk Management. Seven out of the ten cases examined believed Risk Management to be a specialized skill set that should be reserved for IA and not included within CS. The core argument is Risk Management is way too broad of a topic to be covered within CS. Perhaps basic concepts within Risk Management could be covered within an introductory class, along with Security Architecture and Incident Handling. To truly grasp Risk Management, various classes would need to be presented. This would take too much time and effort away from the core concepts that need to be covered within a CS program. The proposal is to not include Risk Management as a concept that should be covered within the CS curriculum.

Privacy

Privacy is another topic, like Ethics, that was considered a mundane topic. Of the cases that were examined, Privacy was perfectly split in half: with five cases believing Privacy was a topic that should be covered within IA, while the other five cases believed Privacy should be covered within CS. Within the five cases that felt Privacy should be taught within CS, there was no consensus on how it should be taught. Two cases believed it should be taught as a standalone class, while two cases felt is should be integrated throughout CS. The final case felt Privacy should be both a dedicated class and integrated throughout a CS program. There really is no clear answer on how or if Privacy should be included within CS.

CS majors need to have an understanding of the need for Privacy. Systems they design and build more than likely will be storing various data points that are regulated. Ideas like tokenization need to be covered. If this was solely in an introductory class the concept of Privacy would be lost. Teaching the concept of Privacy only to IA majors is identical to the concept of including security at the end. CS majors need to know how to build systems that incorporate concepts of Privacy. The proposal is to integrate Privacy throughout a CS curriculum.

Proposition Two: Software Engineering

The literature review defined several IA concepts that should be addressed throughout the SE curriculum. They are: buffer overflows; SQL injection attacks; cross-site scripting; cross-site request forgery; click jacking of code; broken session authentication and session management; insecure direct object references; security misconfigurations; insecure cryptographic storage; failure to restrict URL access; insufficient transport layer protection; and invalidated redirects and forwards. The literature review also defined two core IA concepts that should be included in SE. They are: cryptography and software risk and project management.

Common Vulnerabilities

It is no surprise that nine out of the ten cases examined believe teaching Common Vulnerabilities needs to be included within SE. Eight of the nine cases firmly believe integrating Common Vulnerabilities throughout SE is a requirement. The other case felt Common Vulnerabilities should be covered in an introductory class, but could also be integrated throughout a SE program as well.

Only a single case felt it belonged solely within IA. Case 2 (2013) argued that different programming languages address Common Vulnerabilities differently. Students may graduate knowing a couple of core development languages, but realistically would probably develop in a different language once hired by an organization, or overtime may switch to a newer development language. Case 2 (2013) felt it was the responsibility of the hiring organization to teach Common Vulnerabilities and keep their workforce current on Common Vulnerabilities. An interesting argument, but most of the examined cases agree Common Vulnerabilities should be taught throughout SE, the semantics may change, but the vulnerability concepts have not changed over the past few decades. The proposal is to integrate the concepts of common vulnerabilities throughout the SE curriculum.

The findings from this research support this recommendation, and were what was to be expected. The concept of Common Vulnerabilities is a relatively simple concept to incorporate within SE. SE majors is constantly developing code in homework and labs. These labs should include examples of common vulnerabilities and the ease of exploiting these vulnerabilities.

Cryptography

Cryptography overwhelmingly is supported by a majority of the cases examined. Only one case thought it belonged within IA. This case scoped cryptography as the actual design and analysis of cryptography. Not the use of cryptography algorithms within the code, such as encrypted cookies, SSL transport layer security, and other various important topics.

The other nine cases examined were split on how to incorporate cryptography within SE. Five of the nine cases believed cryptography should be integrated throughout the SE program, while four of the cases felt cryptography should be a standalone class within SE. Unlike CS, where an introductory class could encompass various IA topics, an introductory class in cryptography is not proposed. Like common vulnerabilities, cryptography or the use of cryptography should be included in all classes as a default storage and transport layer security. Coding examples should also include secure mechanisms where possible. An optional standalone class could be presented that analyzes cryptographic algorithms.

Software Risk and Project Management

The need for Software Risk and Project Management within SE is supported by the cases that were analyzed. Only two cases felt that Software Risk and Project Management should be taught solely within IA. Both cases examined believed that Software Risk was a specialty skill set and it was too much to ask a software engineer to remember. Both felt it was the responsibility of specialized security experts to practice Software Risk Management. Both cases also felt that Project management should be de-coupled from Software Risk Management.

The other eight cases examined were perfectly split; four arguing for integration and the other four arguing for a dedicated class within SE. This is a tough topic to determine the correct methodology for incorporation. It could be seen as a topic similar to Risk Management within CS. If that approach were to be taken, it would be exclusively taught within IA. Like Risk Management, the topic of Software Risk and Project Management is a bit broad and would require multiple classes to cover each topic within

the scope of Software Risk and Project Management. By integrating Software Risk and Project Management within SE, concepts could be spread out and re-iterated throughout the program, and therefore enforce retention of key Software Risk and Project Management. Therefore the proposal for Software Risk and Project Management would be to integrate it through several classes within SE.

Proposition Three: Information Technology

The literature review defined several core IA concepts that are combined to create a set of classes specific to IA within IT. These core IA concepts are: fundamental aspects; security mechanisms and countermeasures; operational issues; policy; attacks; security domains; forensics; information states; security services; threat analysis models; and vulnerabilities.

Fundamental Aspects of Information Assurance

All cases agreed that fundamental aspects of information assurance should be taught within IT. The methodology of teaching fundamental aspects is not clear. Exactly half of the cases examined felt it should be integrated throughout IT, while the other half of classes believed it should be a dedicated class within IT. Further analysis with more cases or a survey will need to be completed to better understand which methodology would work better. Perhaps a hybrid solution makes sense. Three cases felt that a fundamentals class should be a mandatory class at the beginning of the IT program. The proposal is to first present these concepts in an introductory class, then reinforce these concepts by integrating them throughout an IT curriculum. Using the definition of Fundamental Aspect if Information Assurance from Machonachy, Schou, Ragsdale, and Welch (2001), it should be relatively easy to integrate the idea of security services

(availability, integrity, authentication, confidentiality and non-repudiation), security countermeasures (technology, policy, and people), and information states (transmission, storage, and processing). Various regulations and governance require these concepts to be included in IT infrastructures. Case studies are an excellent example of how Fundamental Aspects of information assurance could be added to an introductory class or through various courses. This will enable students to retain more IA concepts and apply them when they enter the workforce. .

Security Mechanisms and Countermeasures

Seven of the ten cases believed Security Mechanism and Countermeasures should be included within the IT curriculum. The problem cited by many of the cases was the scope of the definition of Security Mechanisms and Countermeasures. The scope seemed too broad for many of the cases, so no real consensus on how Security Mechanisms and Countermeasures should be included within IT. The ACM and IEEE (2008b) definition was given to each case, which states the following topics: cryptography, authentication, redundancy, and intrusion detection.

The first proposal within Security Mechanisms and Countermeasures is to teach Cryptography as a specialized class within IT. All students graduating with an IT degree need to know the various forms of encryption and when to apply cryptography. For example, a student needs to know how to protect data at rest through the use of cryptography and know when to use database encryption verses full disk encryption. Students also need to know how to encrypt the transmission of data and the importance of encrypting the transmission of data. A lab around SSL Strip would be a great hands-on

example for the need of always encrypting all data transmission, no matter where that data transmission is occurring.

The second proposal within Security Mechanisms and Countermeasures is to include intrusion detection as a concept within the network security class. All IT students need to understand that at some point they, or the organization they work, for will be compromised. Only implementing preventative controls is no longer a valid option for any organization. IT professionals need to also be able to detect a compromise, such as intrusion detection by using the network infrastructure as a holistic sensor. Traditional IT programs taught students how to configure an IDS / IPS appliance. This is no longer acceptable. Students need to understand deep packet analysis and collecting meta-data with netflow analysis. Network labs can easily be setup to walk through various security scenarios that would teach the student more than how to configure an IDS / IPS appliance.

The final proposal within Security Mechanisms and Countermeasures is to integrate the remaining concepts throughout the rest of the IT curriculum. Concepts like authentication, authorization, redundancy can easily be included in various labs and homework assignments asking a student to design a system which is highly redundant, or a system which needs to guarantee non-repudiation.

Operational Security

Operational security viewpoints varied greatly. The definition of Operational Security given to each case was the definition defined by Baird and Gamble (2010), who state Security Operations is "event monitoring; access control; incident investigation; and policy enforcement. Seven of the ten cases felt it should be included within IT, with five

of the seven cases believing it should be integrated throughout the IT curriculum. One of the two cases that argued for a standalone class felt that Operational Security should be an introductory class, and then perhaps integrated throughout. Unfortunately the better solution presented is the lone outlier case. Case 9 believed a dedicated class could be established which would be the ideal scenario to introduce students to red team and blue team exercises.

The concern of developing a red team / blue team exercise is if the student will have enough expertise in Operational Security to make it a worthwhile exercise. Some level of expertise is required by each of the students. A red team / blue team exercise could divide the students based on his or her expertise. For example, a team could consist of a Linux expert, a network expert, and a Microsoft OS expert. Each team member would rely on all of the other team members to get them through the exercise. If an expertise was missing, the instructor could step in and provide help to get the team moving and back on track. The use of a red team / blue team would be an excellent holistic exercise that would enforce the learning of event monitoring, access control, investigation, and policy enforcement.

Policy Creation and Management

Eight of the ten cases examined agreed Policy Creation and Management was a significant topic to be taught within an IT curriculum. Six of the eight cases also agreed that Policy Creation and Management is a significant topic to be a standalone class within the IT curriculum. It should be added that one case felt that Policy Creation and Management belonged in IA. They did feel that Policy Creation and Management could be added as an elective class within IT, but should not be required. This further supports

the need for a dedicated class within an IT curriculum. The proposal is to include Policy Creation and Management as a standalone class within IT.

Although a mundane topic, the need for Policy Creation and Management is an important one. The outcome from the studied cases is what was expected. Policies are typically associated with human resources or information security, but policies are also important for IT. Policies can be compared to the constitution. Policies, much like the constitution define the underlying framework in which statutes are defined. In this case the standards, which define how systems are built.

Attacks

All ten cases believed the concept of attacks should be included within an IT curriculum. This outcome was what was expected. Seven of the ten cases agreed that the concepts of attacks should be integrated throughout the IT curriculum. One case suggested not only having attacks integrated throughout the IT curriculum, but also including a dedicated introductory course at the beginning of the curriculum, to show the students how little or how much they already know; then an advanced class at the end to show the students how much they have learned over the entire program. This is an interesting idea, but timelines are already tight and adding classes to gage progress is typically not an option.

The proposal is to integrate the concepts of attacks throughout the IT curriculum. Including all the different types of attacks within a single class would not provide enough repetition for a student to remember. The hope is that repeating the concept of attacks through several IT courses will reinforce the understanding and application of attack measures and countermeasures.

Security Domains

The concept of security domains within IT was agreed upon by all cases except one. Case 8 (2013) felt security domains belonged in IA. When asked to elaborate on his or her perception, they could not. The issue with security domains is how to incorporate it within IT. Four of the nine cases believed security domains should be integrated throughout the IT curriculum, while the other five cases felt it should be a standalone class within IT.

The definition of Security Domains was taken from the ACM and IEEE (2008b) definition which defines Security Domains as: human-computer interaction, information management, integrative programming, networking, program fundamentals, platform technologies, system administration, system integration / architecture, social issues, web systems and physical plant. The proposed solution would be to integrate security domains into an introductory class, but also integrate security domains throughout the IT curriculum. Since several Security Domains overlap with core IT curriculum, integrating Security Domains within IT should be relatively easy. This will allow students to be exposed to the concept of security domains various times throughout the IT program, while emphasizing the importance of security domains in IT.

Forensics

The results for Forensics were identical to the results within CS. Seven of the ten cases (see Appendix B) examined believed Forensics was too specialized and should be taught solely within IA. Two of cases felt Forensics should be a standalone class within IT, while only one case believed it should be integrated within the IT curriculum. Even

that single case felt teaching Forensics in an undergraduate IT program would be pushing the limits of what could be learned.

Much like the perception to exclude Forensics from within CS, from the researcher's personal observation, this was the result to be expected. The researcher teaches various IT undergraduate classes, of which a set of computer forensics classes is included. The researcher noticed that many students did not have the required skill sets to complete most, if not all of the required labs. Simple skills like command line parameters were not known or not understood. Labs over time evolved and became simpler, either through pre-scripting various commands or documenting step-by-step instructions. This unfortunately can cause students to miss critical thinking skills and rely on canned scripts or perfect textbook scenarios and shows students are not ready to fully understand the different concepts within Forensics. The proposal would be to drop Forensics from the IT curriculum in favor of included it within an IA curriculum as a mandatory core competency.

Information States

The consensus is Information States should be included as a concept within the IT curriculum. Seven out of the ten cases reviewed believed Information States should be integrated throughout the IT curriculum. The proposal is to integrate the concepts of Information States throughout existing defined classes within IT curriculum. This is probably being done today, just not specifically called out as a learning objective.

Security Services

Six of the ten evaluated cases (see Appendix B) agreed that Security Services should be included as a standalone class within IT, but several of the cases believed it

should be part of an introductory level class rather than a dedicated class on its own. Each case was given the following definition by NIST (2003) for Security Services: management (security program, security policy, risk management, security architecture, certification / accreditation, and evaluation), operational (contingency planning, incident handling, testing, and training), and technical (firewalls, intrusion detection, and PKI). Several of these concepts were repeated through various definitions. Those concepts, like security policy, risk management, security architecture, and security operations were examined as independent topics. The proposal is to include the various aspects of Security Services as an introductory level class at the beginning of the IT curriculum.

Threat Analysis Models

There really is no clear answer to where Threat Analysis should be incorporated. Based solely on the number of cases, six cases agree Threat Analysis should be included within the IT curriculum. Only four of the six cases (see Appendix B) believed it should be integrated throughout the IT curriculum, while the other two cases felt Threat Analysis warranted being a separate class within IT. The proposals is to include Threat Analysis as a concept covered within an introductory class in IT, then integrate basic concepts as examples in existing offered IT classes. This supports the need to cover Threat Analysis within an IT curriculum, but keep the understanding at a high level and offer an in-depth class within IA for students who wish to thoroughly understand Threat Analysis. By first covering threats as a basic concept, student will be able to differentiate threats, vulnerabilities and attacks. Adding threats to other existing IT classes puts the concepts of threats in context with specific IT functions, thus strengthening the overall understanding of Threat Analysis.

Vulnerabilities

The final concept examined was Vulnerabilities. Eight of the ten cases (See Appendix B) agreed that Vulnerabilities needed to be taught within the IT curriculum. This outcome is what was to be expected. Six of the eight cases (see Appendix B) believed that Vulnerabilities should be integrated throughout the IT curriculum, while the two other cases felt it should be a standalone class within IT. The proposal is to integrate the concept of Vulnerabilities throughout the existing IT curriculum. The concept of Vulnerabilities can easily be incorporated into case studies, labs, and homework assignments. Being repeating through a multitude of classes will help enforce the fundamental vulnerabilities that have existed since the beginning of IT as a discipline.

Implications to Computer Science and Information Technology

The implications to CS and IT programs will mean some fundamental changes to classes being offered by colleges and universities. Each case was allowed to elaborate on any other comments or suggestions he or she may have on IA. Five core themes were developed from these open discussions. They are: General knowledge; Cross discipline; Critical thinking; Apprentice / mentorship; and Historical perspective. Of these themes, all but one (Cross discipline) has a direct impact on the class structure and layout within CS.

General knowledge is mentioned by several cases (Case 1, Case 3, Case 4, Case 8, Case 9, & Case 10, 2013) as a shortcoming within higher education. The security experts today learned on the job, and came from various backgrounds such as network or system administrators. Examining the backgrounds of the cases analyzed supports this. None of the cases examined had a formal CS or IT degree. Their experiences came from on the

job learning. CS and IT programs today can offer this general knowledge background, but the application of this knowledge is weak, which leads to the second theme of an apprentice / mentorship.

Several of the cases (Case 3, Case 5, Case 9, & Case 10, 2013) felt all colleges and universities should require some sort of apprentice or mentorship program that would allow students to apply what they are learning and enforce the concepts of IA in everyday practice. This will give students the chance to practice some, if not all of the general knowledge they have acquired through repetition and practice.

Critical thinking is another common theme analyzed by several cases (Case 2, Case 4, & Case 7, 2013). Every technical field is changing on a daily basis. What a student knows today will probably be replaced in the near future with another concept, which is perhaps more effective or more efficient. Students need to be able to think and apply what they have learned to various scenarios, and in many cases need to understand when they need to change his or her way of thinking to address some new technology or problem. This can be challenging in a classroom environment, but can be addressed in an apprentice / mentorship program. This will again allow students to learn through repetitive practice and critical thinking; not through learning by memorization or walking through a textbook lab scenario step by step.

The final theme is to not forget a historical perspective. Two of the ten examined cases (Case 5 & Case 10, 2013) believed we are doomed to repeat ourselves if we are not taught our past. Ironically, the field of computer science was developed out of a need for information security. A simple introductory class in CS around the historical perspective could easily be introduced into the curriculum. This historical class could touch on

various IA concepts and the changes that needed to be made in history to address some of the CS and IT shortcomings.

Implications to Software Engineering

The implications of this research could have a major impact within computer science and information technology. The integration of IA concepts within software engineering should have little impact. No specialty classes are required or need to be removed from within SE. Only labs, coding examples, etc. would need to be updated within the SE curriculum.

Recommendations

Computer Science

This research is recommending a substantial change to what IA concepts should be integrated or offered as a standalone class within a computer science curriculum. Several IA concepts: incident handling, security architecture, and privacy, are considered important concepts that should be understood, but separate dedicated classes or the repetition of the concept throughout the curriculum was not warranted. An introductory class containing these concepts should be created and offered at the beginning of the curriculum.

Cryptography is the only IA concept that will be included not only as a standalone class within a CS curriculum, but also integrated throughout the CS curriculum. This integration could be done in case studies, lab exercises, or examples provided within the CS coursework. Ethics was the only other concept that many felt should be integrated throughout the CS curriculum. Having ethics as a standalone class was considered too concentrated and mundane for students to want to learn or comprehend in a single class.

The importance of Ethics warrants it to be repeated through several classes and not included in a single introductory class.

Network Security is the only IA concept that will be taught as a standalone class within the CS curriculum. The perception is Network Security is a core IA concept that contains many sub-concepts. Integrating all these sub-concepts throughout a CS curriculum would weaken the overall core concept of Network Security. Teaching Network Security as part of an introductory class would not cover the concept in enough detail for a student to fully understand or comprehend.

Finally the concepts of Risk Management and Forensics will be completely removed from the CS curriculum. Perhaps the most obstructive is the removal of Forensics from the CS curriculum. Today, Forensics is a popular IA concept taught in many CS curricula. Unfortunately the popularity of Forensics due to television shows and movies will make this difficult for many colleges and universities to want to remove Forensics from a CS curriculum. Risk Management on the other hand is not as popular as Forensics within CS. Removing Risk Management concepts from a CS curriculum will have little impact.

Software Engineering

This research recommends a minimal change to IA concepts that will be integrated or offered within a software engineering curriculum. The recommendation is to integrate the IA concepts of common vulnerabilities, cryptography, and software risk management / project management throughout the SE curriculum. This will be accomplished through lab exercises, case studies, and examples covered through the various classes that comprise the SE curriculum.

Information Technology

This research recommends that the IA concepts: fundamental aspects of security; operational security; attacks; information states; and vulnerabilities be integrated throughout an information technology curriculum. Like CS and SE, this can be accomplished through examples, lab exercises, case studies, etc. Specific topics can be addressed in related classes, such as vulnerabilities in network systems.

The IA concepts of Security Services and Threats are recommended to be included within the IT curriculum as part of an introductory class. Security Domains is an IA concept that should be included as both an introductory class and also integrated throughout the IT curriculum. The various concepts within Security Domains are important enough to be included as an introductory class, but also need to be repeated throughout the IT curriculum in order for students to retain these important concepts.

Security Mechanisms and Countermeasures are believed to incorporate too many IA sub-concepts. The recommendation is to break up Security Mechanisms and Countermeasures into three core sections. The first section is to break off cryptography into a standalone class within the IT curriculum. The second section would be to break off the topic of Intrusion Detection and incorporate it into an existing networking class. The final section would be to integrate the remaining sub-concepts of authentication and redundancy into the IT Curriculum.

The final recommendation is to completely remove the IA concept of Forensics from the IT curriculum. The concept of Forensics is an advanced topic that requires mastery of all IT topics before it can be taught and fully understood. Teaching Forensics

without the core rudimentary IT concepts in place would diminish the value and overall understanding of any Forensics concepts being presented

Further Research

This research did uncover several gaps in research and exploration. One gap is the examination of actual computer science, software engineering, and information technology programs within colleges and universities. The initial proposal for this research included an analysis of current programs by examining the syllabi for all of the courses within the curriculum. This was removed to tighten the scope and concentrate on what should be taught. Examining the current inclusion of IA concepts within each field of study would help establish a baseline comparison of information assurance concepts within each program. A more in depth interview with various colleges and universities could provide even more insight into what is being taught, rather than what IA concepts should be taught within CS, SE, and IT.

A second gap would be to examine colleges and universities which offer a CS, SE, or IT undergraduate degree along with an IA undergraduate degree. It would be interesting to see if there is a correlation between colleges and universities that offer IA programs, and the inclusion of IA concepts within the other undergraduate programs they offer. Do colleges and universities that offer both programs also share instructors and professors? Does this lead to the sharing some common IA concepts as well?

A third gap would be to convert this study into a quantitative research and modify the research questions into an online survey. Each interview question wording could be changed to incorporate a Likert scale to capture a quantitative measurement. The survey could then be distributed to a larger population, perhaps any IA practitioner and

not just organizational leaders. This would further help generalize the perceptions of the IA community.

A final gap that could be explored is the need for IA as an undergraduate program. Several cases that were examined suggested the basic IA concepts should be included within CS, SE, or IT, but a dedicated IA undergraduate program was premature. The perception was IA as a program should be a graduate program, in which CS, SE, or IT graduates feed into it. This can be compared to the idea of a pre-med degree or a degree in architecture. Many states require a Masters degree in order to be certified in Architecture. IA practitioners are similar to architects. An architect needs to have a thorough knowledge in plumbing, electrical, structural engineering, and other infrastructure requirements. Similarly, an IA practitioner needs to have an expert knowledge in networking, system administration and other infrastructure components. Does it make sense to only teach IA as a graduate level program?

REFERENCES

Abi-Antoun, M., & Barnes, J. M. (2010). Analyzing security architectures. In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, ASE 2010.* New York, NY: ACM. 3-12. http://doi.acm.org/10.1145/1858996.1859001

Accreditation Board for Engineering and Technology. (2012). Criteria for accrediting computing programs, 2012-2013. Retrieved from http://www.abet.org/computing-criteria-2012-2013

Association for Computing and Machinery. (2009). Retrieved from http://www.acm.org/about/history

Association for Computing and Machinery, & Institute of Electrical and Electronics Engineers Computer Society. (2004). *Software engineering 2004: Curriculum guidelines for undergraduate degree programs in software engineering*. Retrieved from http://sites.computer.org/ccse/SE2004Volume.pdf

Association for Computing and Machinery, & Institute of Electrical and Electronics Engineers Computer Society. (2008a). *Computer science curriculum 2008: An interim revision of CS 2001*. Retrieved from http://www.acm.org//education/curricula/ComputerScience2008.pdf

Association for Computing and Machinery, & Institute of Electrical and Electronics Engineers Computer Society. (2008b). *Information technology 2008: Curriculum guidelines for undergraduate degree programs in information technology*. Retrieved on from http://www.acm.org/education/curricula/IT2008%20Curriculum.pdf

Al-Hamdani, W. A. (2006). Knowledge flow with information assurance track. In *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development, InfoSecCD 2006*. New York, NY: ACM. 52-57. http://doi.acm.org/10.1145/1231047.1231058

Amer, S. H., & Hamilton, J. A. (2008). Understanding security architecture. In *Proceedings of the 2008 Spring Simulation Multi-conference, SpringSim 2008.* San Diego, CA: Society for Computer Simulation International. 335-342.

Artner, B. (2001). *The importance of the repetition imperative*. Retrieved from http://articles.techrepublic.com.com/5100-10878_11-1039179.html

Attewell, P. (1992, February). Technology diffusion and organizational learning: The case of business computing. *Organization Science*, *3*(1), 1-19.

Baird, R., & Gamble, R. (2010). Reasoning about policy noncompliance. In F. T. Sheldon, S. Prowell, R. K. Abercrombie, & A. Krings (Eds.), *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, CSIIRW 2010*, Oak Ridge, Tennessee, April 2010. New York, NY: ACM. 1-4. http://doi.acm.org/10.1145/1852666.1852736

Bayne, J. (2002). *An overview of threat and risk assessment*. Retrieved from http://www.sans.org/reading-room/whitepapers/auditing/overview-threat-risk-assessment-76

Bellovin, S. M., Benzel, T. V., Blakley, B., Denning, D. E., Diffie, W., Epstein, J., & Verissimo, P. (2008). Information assurance technology forecast 2008. *IEEE Security and Privacy 6*(1), 16-23. http://dx.doi.org/10.1109/MSP.2008.13

Benamati, J. H., Ozdemir, Z. D., & Smith, J. H. (2010). Aligning undergraduate was curricula with industry needs. *Communications of the ACM, 53*(3), 152-156. http://doi.acm.org/10.1145/1666420.1666458

Bhagyavati, M. O., Naugler, D., & Frank, C.E. (2005). Information assurance in the undergraduate curriculum. In *Proceedings of the 43rd Annual Southeast Regional Conference, ACM-SE 43*, New York, NY: ACM, 25-26. http://doi.acm.org/10.1145/1167350.1167368

Bishop, M., & Frincke, D. A. (2008). Information assurance education: A work In progress. *IEEE Security and Privacy 6*(5), 54-57. http://dx.doi.org/10.1109/MSP.2008.123

Blackwell, C. (2009). A security architecture to protect against the insider threat from damage, fraud and theft. In F. T. Sheldon, G. Peterson, A. Krings, R. Abercrombie, & A. Mili (Eds.), *Proceedings of the 5th annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, CSIIRW 2009*, Oak Ridge, Tennessee, April 2009. New York, NY: ACM. 1-4. http://doi.acm.org.library.capella.edu/10.1145/1558607.1558659

Bloom, B.S., & Krathwohl, D. R. (1956) *Taxonomy of educational objectives: The classification of educational goals.* New York, NY: Longmans, Green & Co.

Blyth, A., & Kovacich, G. L. (2001). *Information assurance: Security in the information environment*. New York, NY: Springer.

Bogolea, B., & Wijekumar, K. (2004). Information security curriculum creation: A case study. In *Proceedings of the 1st Annual Conference on Information Security Curriculum Development, InfoSecCD 2004*, Kennesaw, Georgia, October 2004. New York, NY: ACM. 59-65.

Botta, D., Werlinger, R., Gagne, A., Beznosov, K., Iverson, L., Fels, S., & Fisher, B. (2007). Towards understanding IT security professionals and their tools. In *Proceedings of the 3rd Symposium on Usable Privacy and Security, SOUPS 2007*. New York, NY: ACM. 100-111. http://doi.acm.org/10.1145/1280680.1280693

Bratus, S., Shubina, A., & Locasto, M. E. (2010). Teaching the principles of the hacker curriculum to undergraduates. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education, SIGCSE 2010*. New York, NY: ACM. 122-126. http://doi.acm.org/10.1145/1734263.1734303

Brewer, D. C. (2005). *Security controls for Sarbanes-Oxley section 404 IT compliance: authorization, authentication, and access*. Hoboken, NJ: John Wiley and Sons.

Bruschi, D., De Win, B., & Monga, M. (2006). Software engineering for secure systems. In *Proceedings of the 28th International Conference on Software Engineering, ICSE 2006*, Shanghai, China, May 20-28, 2006. New York, NY: ACM. 1007-1008. http://doi.acm.org/10.1145/1134285.1134476

Brynielsson, J. (2009). An information assurance curriculum for commanding officers using hands-on experiments. *Special Interest Group Computer Science Education Bulletin, 41*(1), 236-240. http://doi.acm.org/10.1145/1539024.1508953

Centers of Excellence in Information Assurance Education. (2012). Retrieved from http://www.nsa.gov/ia/academic_outreach/nat_cae/cae_iae_program_criteria.shtml#1

Cannoy, S. D. & Salam, A. F. (2010). A framework for health care information assurance policy and compliance. *Communication of the ACM, 53*(3), 126-131. http://doi.acm.org/10.1145/1666420.1666453

Carlin, A., & Gallegos, F. (2007). IT audit: A critical business process. *Computer, 40*(7), 87-89. http://dx.doi.org/10.1109/MC.2007.246

Cate, F. H. (2009). Security, privacy, and the role of law. *IEEE Security and Privacy, 7*(5), 60-63. DOI= http://dx.doi.org/10.1109/MSP.2009.135

Chatmon, C., Chi, H., & Davis, W. (2010). Active learning approaches to teaching information assurance. In *2010 Information Security Curriculum Development Conference, InfoSecCD 2010*. New York, NY: ACM. 1-7. http://doi.acm.org/10.1145/1940941.1940943

Chess, B. (2002). Improving computer security using extended static checking. In *Proceedings of the IEEE Symposium on Security and Privacy*. Oakland, CA: IEEE Computer Society Press. 118-130.

Chow, T., Chmura, A., & Linberg, K. (2007). Curriculum design for a PhD specialization in IT education. In *Proceedings of the 8th ACM SIGITE Conference on Information Technology Education, SIGITE 2007.* New York, NY: ACM. 21-26. http://doi.acm.org/10.1145/1324302.1324308

Clark, K., Singleton, E., Tyree, S., & Hale, J. (2008). Strata-Gem: risk assessment through mission modeling. In *Proceedings of the 4th ACM Workshop on Quality of Protection, QOP 2008.* New York, NY: ACM. 51-58. http://doi.acm.org/10.1145/1456362.1456374

Committee on National Security Systems. (2004a). *National information assurance training standard for senior system managers*. Retrieved from http://www.cnss.gov/Assets/pdf/cnssi_4012.pdf

Committee on National Security Systems. (2004b). *National information assurance training standard for system administrators (SA).* Retrieved from http://www.cnss.gov/Assets/pdf/cnssi_4013.pdf

Committee on National Security Systems. (2004c). *National information assurance training standard for information systems security officers.* Retrieved from http://www.cnss.gov/Assets/pdf/cnssi_4014.pdf

Committee on National Security Systems. (2005). *National information assurance training standard for risk analysts.* Retrieved from http://www.cnss.gov/Assets/pdf/CNSSI-4016.PDF

Committee on National Security Systems. (2006). *National information assurance glossary.* Retrieved from http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf

Conti, G., Hill, J. M. D., Lathrop, S., Alford, K., & Ragsdale, D. J. (2003). *Towards a comprehensive undergraduate information assurance program.* C. Irvine, & H. Armstrong (Eds.). Norwell, MA: Kluwer Academic Publishers.

Conklin, W. A., & Dietrich, G. (2008). Systems theory model for information security. In *Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences, HICSS 2008.* Washington, DC, USA: IEEE Computer Society. 265-266. http://dx.doi.org/10.1109/HICSS.2008.421

Cooper, P. (2005). Speciation in the computing sciences: digital forensics as an emerging academic discipline. In *Proceedings of the 2nd Annual Conference on Information Security Curriculum Development, InfoSecCD 2005*, Kennesaw, Georgia, September 23 - 24, 2005. New York, NY: ACM. 19-23. http://doi.acm.org/10.1145/1107622.1107628

Cooper, P., Finley, G. T., & Kaskenpalo. P. (2010). Towards standards in digital forensics education. In A. Clear, & L. Russell Dag (Eds.), *Proceedings of the 2010 Innovation and Technology in Computer Science Education Working Group Reports, ITICSE-WGR 2010.* New York, NY: ACM. 87-95. http://doi.acm.org/10.1145/1971681.1971688

Cooper, S., Nickell, C., Piotrowski, V., Oldfield, B., Abdallah, A., Bishop, … Brynielsson, J. (2010). An exploration of the current state of information assurance education. *Special Interest Group Computer Science Education Bulletin, 41*(4), 109-125. http://doi.acm.org/10.1145/1709424.1709457

Creswell, J. W. (1998). *Research design: Qualitative and quantitative approaches.* Thousand Oaks, CA: Sage Publications.

Creswell, J. W. (2009). *Research design: Qualitative, quantitative, and mixed methods approaches.* (3rd ed.). Thousand Oaks, CA: Sage Publications.

Crowley, E. (2003). Information system security curricula development. In *Proceedings of the 4th Conference on Information Technology Curriculum, CITC4 2003.* New York, NY: ACM. 249-255. http://doi.acm.org/10.1145/947121.947178

Crowley, E. (2007). Corporate forensics class design with open source tools and live CDS. *Journal of Computing in Small Colleges, 22*(4), 170-176.

Dahlberg, T., Barnes, T., Buch, K., & Rorrer. A. (2011). The STARS alliance: Viable strategies for broadening participation in computing. *Transactions on Computing Education, 11*(3), 1-25. http://doi.acm.org/10.1145/2037276.2037282

Dark, M. J. (2004) Civic responsibility and information security: An information security management service learning course. In *Information Security Curriculum Development, INFOSECCD 2004.* Retrieved from https://www.cerias.purdue.edu/assets/pdf/bibtex_archive/2004-43.pdf

Dark, M. J., Ekstrom, J. J., & Lunt, B. M. (2005). Integration of information assurance and security into the IT2005 model curriculum. In *Proceedings of the 6th Conference on Information Technology Education, SIGITE 2005.* New York, NY: ACM. 7-14. http://doi.acm.org/10.1145/1095714.1095719

Dimkov, T., Pieters, W., & Hartel, P. (2011). Training students to steal: a practical assignment in computer security education. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, SIGCSE 2011.* New York, NY: ACM. 21-26. http://doi.acm.org/10.1145/1953163.1953175

Dottore, A. G. (2009). Business model adaptation as a dynamic capability: a theoretical lens for observing practitioner behavior. *22nd Bled eConference*, June 14-17, 2004. Retrieved from https://domino.fov.uni-mb.si/proceedings.nsf/Proceedings/AFF4E9411E5A8229C12576000040513E/$File/32_Dottore.pdf

Edge, C., & Stamey, J. (2010). Security education on a budget: getting the most "bang for the buck" with limited time and resources. In *2010 Information Security Curriculum Development Conference, InfoSecCD 2010.* New York, NY: ACM. 29-35. http://doi.acm.org/10.1145/1940941.1940949

Fass, L F. (2008). An ethnocentric look at the law and technology interface. *Special Interest Group on Software Engineering Notes 33*(1), 1-10. http://doi.acm.org/10.1145/1344452.1344460

Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography engineering: Design principles and practical applications.* Indianapolis, IN: Wiley.

Ferrara, E. S. (2006). *The need for information assurance in the 21st century.* Retrieved from http://www.esferrara.com/TheRoleofInformationAssuranceinthe21.pdf

Fitzgerald, T. (2008). *Business drivers for information security: Who needs them anyway?* Retrieved from http://www.bloginfosec.com/2008/07/23/business-drivers-for-information-security-who-needs-them-anyway/2

Foley, S. N. (2009). Security risk management using internal controls. In *Proceedings of the First ACM Workshop on Information Security Governance, WISG 2009,* Chicago, Illinois, November 13, 2009. New York, NY: ACM. 59-64. http://doi.acm.org/10.1145/1655168.1655179

Frank, C. E., & Werner, L. (2010). The benefit of the CSSLP certification for educators and professionals. *Journal of Computing in Small Colleges, 26*(1), 49-55.

Frost & Sullivan. (2011). Retrieved from https://www.isc2.org/uploadedFiles/Industry_Resources/FS_WP_ISC%20Study_020811_MLW_Web.pdf

Geoghegan, S. J. (2008). The development of an information assurance program. *Journal of Computing in Small Colleges, 23*(4), 116-123.

Ghafarian, A. (2007). Ideas for projects in undergraduate information assurance and security courses. In *Proceedings of the 12th Annual Special Interest Group: Computer Science Education Conference on Innovation and Technology in Computer Science Education, ITICSE 2007.* New York, NY: ACM. 322-322. http://doi.acm.org/10.1145/1268784.1268889

Gilbert, F.. (2009, July). Seven drivers for privacy and security issues in a down economy. *Journal of Internet Law, 13*(1), 3-7.

Goel, S., Pon, D., Bloniarz, P., Bangert-Drowns, R., Berg, G., Delio, V., Iwan, L., Hurbanek, T., Schuman, S. P., Gangolly, J., Baykal, A., & Hobbs, J. (2006). Innovative model for information assurance curriculum: A teaching hospital. *Journal on Educational Resources in Computing, 6*(3), 2-2.

Gray, S., St. Clair, C., James, R., & Mead, J. (2007). Suggestions for graduated exposure to programming concepts using fading worked examples. In *Proceedings of the Third International Workshop on Computing Education Research, ICER 2007.* New York, NY: ACM. 99-110. http://doi.acm.org/10.1145/1288580.1288594

Greene, R. (1989, May). Spacing effects in memory: Evidence for a two-process account. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*(3), 371-377. doi:10.1037/0278-7393.15.3.371

Gupta, G. K. (2007). Computer science curriculum developments in the 1960s. *IEEE Annals of the History of Computing, 29*(2), 40-54.

Hamilton, J. A., Owor, R. S., & Dajani, K. F. (2009). Building information assurance education partnerships with minority institutions. In *The Fifth Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations, TAPIA 2009.* New York, NY: ACM. 58-63. http://doi.acm.org/10.1145/1565799.1565813

Hjelmås, E., & Wolthusen, S. D. (2006). Full-spectrum information security education: integrating B.Sc., M.Sc., and Ph.D. programs. In *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development, InfoSecCD 2006.* New York, NY: ACM. 5-12.

Humphrey, W. S. (1989). The software engineering process: definition and scope. *Special Interest Group on Software Engineering Notes, 14*(4), 82-83.

IEEE. (2009). Retrieved on from http://www.computer.org/portal/site/ieeecs/menuitem.c5efb9b8ade9096b8a9ca0108bcd45f3/index.jsp?&pName=ieeecs_level1&path=ieeecs/about/history&file=CShistory.xml&xsl=generic.xsl&

Irvine, C.E., Chin, S., & Frincke, D. (1998). Integrating security into the curriculum. *Computer, 31*(12), 25-30. http://dx.doi.org/10.1109/2.735847

Irvine, C., & Nguyen. T. D. (2010). Educating the systems security engineer's apprentice. *IEEE Security and Privacy, 8*(4), 58-61. http://dx.doi.org/10.1109/MSP.2010.123

Islam, S., & Dong, W. (2008). Human factors in software security risk management. In *Proceedings of the First International Workshop on Leadership and Management in Software Architecture, LMSA 2008*, Leipzig, Germany, May 11, 2008. New York, NY: ACM. 13-16. http://doi.acm.org/10.1145/1373307.1373312

Information Technology Association of America. (2008). *Information technology definitions*. Retrieved from http://www.techamerica.org/Docs/fileManager.cfm?f=http://techamerica.org/es/docs/information%20technology%20definitions.pdf

Jensen, B. K., Cline, M., & Guynes, C. S. (2006). Teaching the undergraduate CS information security course. *Special Interest Group on Software Engineering Bulletin, 38*(2), 61-63. http://doi.acm.org/10.1145/1138403.1138434

Kabay, M. E. (2005). Improving information assurance education key to improving secure(ity) management. *Journal of Network and Systems Management, 13*(3), 247-251.

Kamali, R., Liles, S., Winer, C., Jiang, K., & Nicolai, B. (2005). An implementation of the SIGITE model curriculum. In *Proceedings of the 6th Conference on Information Technology Education, SIGITE 2005.* New York, NY: ACM. 15-17. http://doi.acm.org/10.1145/1095714.1095720

Karam, O., & Peltsverger, S. (2009). Teaching with security in mind. In *Proceedings of the 47th Annual Southeast Regional Conference, ACM-SE 47.* New York, NY: ACM. 1-4. http://doi.acm.org/10.1145/1566445.1566536

Knewstubb, B., & Bond, C. (2009, June). What's he talking about? The communicative alignment between a teacher's intentions and students' understandings. *Higher Education Research and Development, 28*(2), 179-193. doi:10.1080/07294360902725058

Lee, J., Bagchi-Sen, S., Rao, H. R., & Upadhyaya, S. J. (2010). Anatomy of the information security workforce. *IT Professional, 12*(1), 14-23. http://dx.doi.org/10.1109/MITP.2010.23

Lee, K., & Mirchandani, D. (2009). Analyzing the dynamics of skill sets for the U.S. information systems workforce using latent growth curve modeling. In *Proceedings of the Special Interest Group on Management Information System's 47th Annual Conference on Computer Personnel Research, SIGMIS CPR 2009.* New York, NY: ACM. 113-120. http://doi.acm.org/10.1145/1542130.1542153

Lester, C. Y. (2010). Shifting the paradigm: Training undergraduate students in software security. In *Proceedings of the 2010 Fourth International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2010.* Washington, DC: IEEE Computer Society. 117-122. http://dx.doi.org/10.1109/SECURWARE.2010.27

Lester, C. Y., & Jamerson, F. (2009). Incorporating software security into an undergraduate software engineering course. In *Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009.* Washington, DC: IEEE Computer Society. 161-166. http://dx.doi.org/10.1109/SECURWARE.2009.32

Lester, C. Y., Narang, H., & Chen, C. (2008). Infusing information assurance into an undergraduate CS curriculum. In *Proceedings of the 2008 Second International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2008.* Washington, DC: IEEE Computer Society. 300-305. http://dx.doi.org/10.1109/SECURWARE.2008.63

Livermore, J., Baker, K., Krolczyk, V., & Saurbier, A. (2011). Capstone, thesis, or practicum?: The state of the practice in IA education. In *Proceedings of the 2011 Information Security Curriculum Development Conference, INFOSECCD 2011.* New York, NY: ACM. 69-73. http://doi.acm.org/10.1145/2047456.2047467

Losavio, M., Shutt, J. E., & Keeling, D. (2010). Positing social and justice models for cyber security. In F. T. Sheldon, S. Prowell, R. K. Abercrombie, & A. Krings (Eds.), *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, CSIIRW 2010.* New York, NY: ACM. 1-4. http://doi.acm.org/10.1145/1852666.1852755

Maconachy, V. W. Schou, C. D., Ragsdale, D., & Welch, D. (2001). A model for information assurance: An integrated approach. *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*. Retrieved from http://www.academia.edu/12837313/A_Model_for_Information_Assurance_An_Integrated_Approach

Maqsood, M., & Javed, T. (2007). Practicum in software project management: An endeavor to effective and pragmatic software project management education. In *Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC-FSE 2007.* New York, NY: ACM. 471-480. http://doi.acm.org/10.1145/1287624.1287691

Markham, S. A. (2009). Expanding security awareness in introductory computer science courses. In *2009 Information Security Curriculum Development Conference, InfoSecCD 2009.* New York, NY: ACM. 27-31. http://doi.acm.org/10.1145/1940976.1940984

Marlinspike, M. (2009) *New tricks for defeating SSL in practice.* Retrieved from https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf

McGuire, T. J., & Murff, K. N. (2006). Issues in the development of a digital forensics curriculum. *Journal on Computing for Small Colleges, 22*(2), 274-280.

Mead, N. R., & Hough, E. D. (2006). Security requirements engineering for software systems: Case studies in support of software engineering education. In *Proceedings of the 19th Conference on Software Engineering Education and Training, CSEET 2006.* Washington, DC: IEEE Computer Society. 149-158. http://dx.doi.org/10.1109/CSEET.2006.30

Mead, N. R., & Jarzombek, J. (2010). Advancing software assurance with public-private collaboration. *Computer, 43*(9), 21-30. http://dx.doi.org/10.1109/MC.2010.247

Meiselwitz, G. H. (2008). Information security across disciplines. In *Proceedings of the 9th ACM SIGITE Conference on Information Technology Education, SIGITE 2008.* New York, NY: ACM. 99-104. http://doi.acm.org/10.1145/1414558.1414588

Microsoft. (2013). Retrieved from https://msdn.microsoft.com/en-us/library/cc751383.aspx

Miller, C. S., & Dettori, L. (2008). Employers' perspectives on it learning outcomes. In *Proceedings of the 9th ACM SIGITE Conference on Information Technology Education, SIGITE 2008.* New York, NY: ACM. 213-218. http://doi.acm.org/10.1145/1414558.1414612

Morneau, K. A. (2004). Designing an information security program as a core competency of network technologists. In *Special Interest Group on Information Technology Education, SIGITE 2004.* doi:10.1145/1029533.1029541

Myers, J. P., & Riela, S. (2008). Taming the diversity of information assurance and security. *Journal of Computing Sciences in Colleges, 23*(4), 173-179.

Neubauer, T., Klemen, M., & Biffl. S. (2005). Business process-based valuation of IT-security. In K. Sullivan (Ed.), *Proceedings of the Seventh International Workshop on Economics-driven Software Engineering Research, EDSER 2005.* New York, NY: ACM. 1-5. http://doi.acm.org/10.1145/1082983.1083099

National Institute of Standards and Technology. (2001). Retrieved from http://nvl.nist.gov/pub/nistpubs/sp958-lide/250-253.pdf

National Institute of Standards and Technology. (2003). *Guide to information technology security services*. (NIST Special Publication No. SP800-35). Gaithersburg, MD: Computer Security Division, Information Technology Laboratory, NIST.

National Institute of Standards and Technology. (2008) *Computer security incident handling guide*. (NIST Special Publication No. SP800-61r1). Gaithersburg, MD: Computer Security Division, Information Technology Laboratory, NIST.

National Security Agency. (2009). Criteria for measurement for CAE/IAE. Retrieved from http://www.nsa.gov/ia/academic_outreach/nat_cae/cae_iae_program_criteria.shtml

National Security Telecommunications and Information Systems Security. (1994). *National training standard for information systems security (INFOSEC) professionals.* Retrieved from http://www.cnss.gov/Assets/pdf/nstissi_4011.pdf

Null, L. (2004). Integrating security across the computer science curriculum. *Journal of Computing in Small Colleges, 19*(5), 170-178.

Open Web Application Security Project. (2007). Retrieved from http://www.owasp.org/index.php/Top_10_2007

Pan, Y. (2007). Security auditing course development. In *Proceedings of the 8th ACM SIGITE Conference on Information Technology Education, SIGITE 2007*. New York, NY: ACM. 259-266. http://doi.acm.org/10.1145/1324302.1324357

Papadopoulos, P., Demetriadis, S., Stamelos, I., & Tsoukalas, I. (2009, April). Prompting students' context-generating cognitive activity in ill-structured domains: Does the prompting mode affect learning? *Educational Technology Research and Development, 57*(2), 193-210. doi:10.1007/s11423-008-9105-6

Park, M. A. (2011). Embedding security into visual programming courses. In *Proceedings of the 2011 Information Security Curriculum Development Conference, InfoSecCD 2011*. New York, NY: ACM. 84-93. http://doi.acm.org/10.1145/2047456.2047469

Payne, J. (2010). Integrating application security into software development. *IT Professional, 12*(2), 6-9. http://dx.doi.org/10.1109/MITP.2010.58

Peltsverger , S., & Karam, O. (2010). Is teaching with security in mind working?. In *2010 Information Security Curriculum Development Conference, INFOSECCD 2010.* New York, NY: ACM. 15-20. http://doi.acm.org/10.1145/1940941.1940946

Peltsverger, S., & Teat, C. (2009). Incorporating current events into information assurance curriculum. In *2009 Information Security Curriculum Development Conference, INFOSECCD 2009.* New York, NY: ACM. 6-9. http://doi.acm.org/10.1145/1940976.1940979

Pérez, L. C., Cooper, S., Hawthorne, E. K., Wetzel, S., Brynielsson, J., Gencer Gokce, A., … Upadhyaya, S. (2011). Information assurance education in two- and four-year institutions. In L. Adams, & J. J. Jurgens (Eds.), *Proceedings of the 16th Annual Conference Reports on Innovation and Technology in Computer Science Education - Working Group Reports, ITICSE-WGR 2011.* New York, NY: ACM. 39-53. http://doi.acm.org/10.1145/2078856.2078860

Peterson, G. (2006). Retrieved from http://arctecgroup.net/pdf/ArctecSecurityArchitectureBlueprint.pdf

Petrova, K., Philpott, A., Kaskenpalo, P., & Buchan, J. (2004). Embedding information security curricula in existing programmes. In *Proceedings of the 1st Annual Conference on Information Security Curriculum Development, INFOSECCD 2004,* Kennesaw, Georgia, October 8, 2004. New York, NY: ACM. 20-29.

Pironti, J. P. (2008). Key elements of an information risk management program: transforming information security into information risk management. *Information Systems Control Journal, 2008*(2), 1-6.

Ponemon Institute. (2014). Retrieved from http://www.hp.com/hpinfo/newsroom/press_kits/2014/RSAConference2014/Ponemon_2014_Best_Schools_Report.pdf

Pothamsetty, V. (2005). Where security education was lacking. In *Proceedings of the 2nd Annual Conference on Information Security Curriculum Development, INFOSECCD 2005.* New York, NY: ACM. 54-58. http://doi.acm.org/10.1145/1107622.1107635

Rao, H. R., Gupta, M., & Upadhyaya, S. (2007*). Managing information assurance in financial services.* Retrieved from http://www.som.buffalo.edu/isinterface/newbook/promtion_toc.pdf

Reid, R. C., & Gilbert, A. H. (2007). Managing security from the perspective of the business executive. In *Proceedings of the 4th Annual Conference on Information Security Curriculum Development, INFOSECCD 2007.* New York, NY: ACM. 1-5. http://doi.acm.org/10.1145/1409908.1409925

Reynolds, C. W., & Goda, B. S. (2007). The affective dimension of pervasive themes in the information technology curriculum. In *Proceedings of the 8th ACM SIGITE Conference on Information Technology Education, SIGITE 2007,* Destin, Florida, October 18-20, 2007. New York, NY: ACM. 13-20. http://doi.acm.org/10.1145/1324302.1324306

Rice, D. (2007). *Geekonomics*. Boston, MA: Addison-Wesley.

Rogers, L. R. (2006). The CERT survivability and information assurance curriculum: Education for first defenders. In *Proceedings of the 10th Colloquium for Information Systems Security Education, CISSE,* Adelphi, MD, June 5-8, 2006. 1-7.

Rowe, D. C., Lunt, B. M., & Ekstrom, J. J. (2011). The role of cyber-security in information technology education. In *Proceedings of the 2011 Conference on Information Technology Education, SIGITE 2011.* New York, NY: ACM. 113-122. http://doi.acm.org/10.1145/2047594.2047628

Rubin, B. S., & Misra, B. S. (2007). Creating a computer security curriculum in a software engineering program. In *Proceedings of the 29th International Conference on Software Engineering, ICSE 2007.* Washington, DC: IEEE Computer Society. 732-735. http://dx.doi.org/10.1109/ICSE.2007.28

System Administration, Networking, and Security Institute. (2012). Retrieved from http://www.sans.org/critical-security-controls/control.php?id=6

Schaefer, R. (2009). The epistemology of computer security. *Special Interest Group on Software Engineering Notes, 34*(6), 8-10. http://doi.acm.org/10.1145/1640162.1655274

Schwarz, T. S. (2005). Teaching ethics and computer forensics: The Markkula center for applied ethics approach. In *Proceedings of the 2nd Annual Conference on Information Security Curriculum Development, INFOSECCD 2005,* Kennesaw, Georgia, September 23-24, 2005. New York, NY: ACM. 66-71. http://doi.acm.org.library.capella.edu/10.1145/1107622.1107637

Schweitzer, D., Gibson, D., & Collins, M. (2009). Active learning in the security classroom. In *Proceedings of the 42nd Hawaii International Conference on System Sciences, HICSS 2009.* Washington, DC: IEEE Computer Society. 1-8. http://dx.doi.org/10.1109/HICSS.2009.47

Schweitzer, D., Humphries, J., & Baird, L. (2006). Meeting the criteria for a Center of Academic Excellence (CAE) in information assurance education. *Journal of Computing for Small Colleges, 22*(1), 151-160.

Sexton, J. (2008). Establishing an undergraduate information assurance (information security) program at a small liberal arts college. *Journal of Computing in Small Colleges, 24*(2), 234-240.

Simmons, C. B., & Simmons, L. L. (2010). Gaps in the computer science curriculum: an exploratory study of industry professionals. *Journal of Computing in Small Colleges, 25*(5), 60-65.

Spears, J. L., & Barki, H. (2010). User participation in information systems security risk management. *MIS Quarterly, 34*(3), 503-522.

Streff K., & Zhou, Z. (2006). Developing and enhancing a computer and network security curriculum. *Journal of Computing in Small Colleges, 21*(3), 4-18.

Taylor, B., & Azadegan, S. (2006). Threading secure coding principles and risk analysis into the undergraduate computer science and information systems curriculum. In *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development, INFOSECCD 2006*, Kennesaw, Georgia, September 22-23, 2006. New York, NY: ACM. 24-29.

Tenenberg, J. (2009). The ultimate guest speaker: a model for educator/practitioner collaboration. *Journal of Computing in Small Colleges, 25*(1), 123-129.

Theoharidou, M., & Gritazalis. D. (2007). Common body of knowledge for information security. *IEEE Security and Privacy, 5*(2), 64-67. http://dx.doi.org/10.1109/MSP.2007.32

Tipton, H. F. (2009). *Official ISC(2) guide to the CISSP CBK*. London, UK: Taylor and Francis.

Tjoa, S., Jakoubi, S., Goluch, G., Kitzler, G., Goluch, S., & Quirchmayr, G. (2011). A formal approach enabling risk-aware business process modeling and simulation. *IEEE Transactions on Services Computing, 4*(2), 153-166. http://dx.doi.org/10.1109/TSC.2010.17

University of Minnesota. (2005). *University Catalogs*. Retrieved from http://onestop2.umn.edu/programCatalog/viewCatalogProgram.do?programID=123&strm=1059

Uzubell, S., Liles, S., & Jiang, K. (2010). An analysis of the common body of knowledge of software assurance. In *Proceedings of the 2010 ACM Conference on Information Technology Education, SIGITE 2010.* New York, NY: ACM. 125-130. http://doi.acm.org/10.1145/1867651.1867684

Vaughn, R., & Dampier, D. (2007). Digital forensics--state of the science and foundational research activity. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences, HICSS.* Washington, DC: IEEE Computer Society. 263. http://dx.doi.org/10.1109/HICSS.2007.174

Walden, J. (2008). Integrating web application security into the IT curriculum. In *Proceedings of the 9th ACM SIGITE Conference on Information Technology Education, SIGITE 2008.* New York, NY: ACM. 187-192. http://doi.acm.org/10.1145/1414558.1414607

Wang, A. J. A. (2008). A security thread in a thread-based curriculum. In *Special Interest Group Information Technology Education, SIGITE 2008.* New York, NY: ACM. 1-2.

Ward, C., Agassi, S., Bhattacharya, K., Biran, O., Cocchiara, R., Factor, M. E., … Wolfsthal, Y. (2009). Toward transforming business continuity services. *IBM Journal of Research and Development, 53*(6), 856-870.

Werlinger, R., Hawkey, K., Botta, D., & Beznosov, K. (2009). Security practitioners in context: Their activities and interactions with other stakeholders within organizations. *International Journal of Human-Computer Studies, 67*(7), 584-606. http://dx.doi.org/10.1016/j.ijhcs.2009.03.002

Whitman, M. E., & Mattord, H. J. (2005). Workshop on designing and teaching information security curriculum. In *Proceedings of the 43rd Annual Southeast Regional Conference, ACM-SE 43.* New York, NY: ACM. 16-17. http://doi.acm.org/10.1145/1167350.1167363

Xiaobin, T., Yong, Z., & Hongsheng, X. (2007). Multi-perspective quantization model for cyberspace security situation awareness. In *Proceedings of the 2007 International Conference on Computational Intelligence and Security, CIS 2007,* Washington, DC: IEEE Computer Society. 853-857. http://dx.doi.org/10.1109/CIS.2007.171

Yang, T. A., & Nguyen, T.A. (2006). Network security development process: a framework for teaching network security courses. *Journal of Computing in Small Colleges, 21*(4), 203-209.

Yin, R. K. (2009). *Case study research: Design and methods* (Vol. 5) (4th ed.). Los Angeles, CA: Sage.

Zambon, E., Bolzoni, D., Etalle, S., & Salvato, M. (2007). A model supporting business continuity auditing and planning in information systems. In *Proceedings of the Second International Conference on Internet Monitoring and Protection, ICIMP 2007*. Washington, DC: IEEE Computer Society. 33. http://dx.doi.org/10.1109/ICIMP.2007.4

APPENDIX A. INSTRUMENTATION

Level 1 Questions:

1) What are your title and roles / responsibilities?
2) What was the highest level of education you have completed?
    a. [Undergraduate] Did you graduate with a Computer Science, Software Engineering, Information Technology, or Other degree

Level 2 Questions

1) Do you think cryptography should be included as a standalone class in Computer Science; integrated into an existing CS class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
2) Do you think forensics should be included as a standalone class in Computer Science; integrated into an existing CS class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
3) Do you think network security should be included as a standalone class in Computer Science; integrated into an existing CS class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
4) Do you think ethics should be included as a standalone class within Computer Science integrated into an existing CS class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
5) Do you think incident handling should be included as a standalone class within Computer Science; integrated into an existing CS class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
6) Do you think security architecture should be included as a standalone class within Computer Science; integrated into an existing CS class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?

b. If not at all why?
c. If a combination of integrated, standalone, or within IA why?

7) Do you think risk management should be included as a standalone class within Computer Science; integrated into an existing CS class; in a dedicated Information Assurance curriculum; or a combination or not at all?
   a. If integrated, standalone or within IA why?
   b. If not at all why?
   c. If a combination of integrated, standalone, or within IA why?
8) Do you think privacy should be included as a standalone class within Computer Science; integrated into an existing CS class; in a dedicated Information Assurance curriculum; or a combination or not at all?
   a. If integrated, standalone or within IA why?
   b. If not at all why?
   c. If a combination of integrated, standalone, or within IA why?
9) Are there any other Information Assurance concepts that were not covered that you feel should be included within Computer Science?
10) Do you think common vulnerabilities (such as code injection, buffer overflows, cross-site scripting, etc) should be integrated throughout a Software Engineering curriculum; integrated into an existing SE class; in a dedicated Information Assurance curriculum; or a combination or not at all?
   a. If integrated, standalone or within IA why?
   b. If not at all why?
   c. If a combination of integrated, standalone, or within IA why?
11) Do you think cryptography should be included as a standalone class within Software Engineering; integrated into an existing SE class; in a dedicated Information Assurance curriculum; or a combination or not at all?
   a. If integrated, standalone or within IA why?
   b. If not at all why?
   c. If a combination of integrated, standalone, or within IA why?
12) Do you think software risk and project management should be included as a standalone class within Software Engineering; integrated into an existing SE class; in a dedicated Information Assurance curriculum; or a combination or not at all?
   a. If integrated, standalone or within IA why?
   b. If not at all why?
   c. If a combination of integrated, standalone, or within IA why?
13) Are there any other Information Assurance that were not covered that you feel should be included within Software Engineering?
14) Do you think fundamental aspects of security should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
   a. If integrated, standalone or within IA why?
   b. If not at all why?
   c. If a combination of integrated, standalone, or within IA why?

15) Do you think security mechanisms and countermeasures should be included as a standalone class within Information Technology; into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
16) Do you think operational security issues should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
17) Do you think policy creation and management should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
18) Do you think attacks should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
19) Do you think security domains should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
20) Do you think forensics should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
21) Do you think information states should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
22) Do you think security services should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?

a. If integrated, standalone or within IA why?
b. If not at all why?
c. If a combination of integrated, standalone, or within IA why?

23) Do you think threat analysis models should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
24) Do you think security vulnerabilities should be included as a standalone class within Information Technology; integrated into an existing IT class; in a dedicated Information Assurance curriculum; or a combination or not at all?
    a. If integrated, standalone or within IA why?
    b. If not at all why?
    c. If a combination of integrated, standalone, or within IA why?
25) Are there any other Information Assurance that were not covered that you feel should be included within Information Technology?

APPENDIX B. RESEARCH POPULATION AND SAMPLING

Table B1

Case 1 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	Standalone
Computer science	Ethics	IA
Computer science	Forensics	IA
Computer science	Incident handling	IA
Computer science	Network security	Standalone
Computer science	Privacy	IA
Computer science	Risk management	IA
Computer science	Security architecture	IA
Information technology	Attacks	Standalone
Information technology	Forensics	Standalone
Information technology	Fundamental aspects	Standalone
Information technology	Information states	Standalone
Information technology	Operational security	Integrated
Information technology	Policy creation and management	Standalone
Information technology	Security domains	Standalone
Information technology	Security mechanisms and countermeasures	Standalone
Information technology	Security services	Standalone
Information technology	Security vulnerabilities	Standalone
Information technology	Threat analysis	Integrated
Software engineering	Common vulnerabilities	Integrated
Software engineering	Cryptography	Standalone
Software engineering	Software risk and project management	IA

Table B2

Case 2 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	Integrated
Computer science	Ethics	Standalone
Computer science	Forensics	IA
Computer science	Incident handling	Integrated
Computer science	Network security	Standalone
Computer science	Privacy	Standalone
Computer science	Risk management	Integrated
Computer science	Security architecture	Standalone
Information technology	Attacks	Integrated
Information technology	Forensics	Standalone
Information technology	Fundamental aspects	Integrated
Information technology	Information states	Integrated
Information technology	Operational security	IA
Information technology	Policy creation and management	IA
Information technology	Security domains	Standalone
Information technology	Security mechanisms and countermeasures	IA
Information technology	Security services	IA
Information technology	Security vulnerabilities	IA
Information technology	Threat analysis	IA
Software engineering	Common vulnerabilities	IA
Software engineering	Cryptography	Integrated
Software engineering	Software risk and project management	Integrated

Table B3

Case 3 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	Integrated
Computer science	Ethics	Standalone
Computer science	Forensics	IA
Computer science	Incident handling	IA
Computer science	Network security	Integrated
Computer science	Privacy	Integrated
Computer science	Risk management	IA
Computer science	Security architecture	Integrated
Information technology	Attacks	Integrated
Information technology	Forensics	IA
Information technology	Fundamental aspects	Integrated
Information technology	Information states	Standalone
Information technology	Operational security	IA
Information technology	Policy creation and management	Standalone
Information technology	Security domains	Integrated
Information technology	Security mechanisms and countermeasures	Integrated / Standalone
Information technology	Security services	Standalone
Information technology	Security vulnerabilities	Integrated
Information technology	Threat analysis	Standalone
Software engineering	Common vulnerabilities	Integrated
Software engineering	Cryptography	Integrated
Software engineering	Software risk and project management	Standalone

Table B4

Case 4 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	Integrated
Computer science	Ethics	Integrated
Computer science	Forensics	Standalone
Computer science	Incident handling	IA
Computer science	Network security	Standalone
Computer science	Privacy	Standalone
Computer science	Risk management	IA
Computer science	Security architecture	Standalone
Information technology	Attacks	Standalone
Information technology	Forensics	IA
Information technology	Fundamental aspects	Standalone
Information technology	Information states	Integrated
Information technology	Operational security	Integrated
Information technology	Policy creation and management	Standalone
Information technology	Security domains	Standalone
Information technology	Security mechanisms and countermeasures	IA
Information technology	Security services	Standalone
Information technology	Security vulnerabilities	IA
Information technology	Threat analysis	IA
Software engineering	Common vulnerabilities	Integrated
Software engineering	Cryptography	Standalone
Software engineering	Software risk and project management	Standalone

Table B5

Case 5 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	Standalone
Computer science	Ethics	Integrated
Computer science	Forensics	Standalone
Computer science	Incident handling	Integrated
Computer science	Network security	Standalone
Computer science	Privacy	Integrated
Computer science	Risk management	IA
Computer science	Security architecture	IA
Information technology	Attacks	Integrated
Information technology	Forensics	IA
Information technology	Fundamental aspects	Integrated
Information technology	Information states	Integrated
Information technology	Operational security	Integrated
Information technology	Policy creation and management	Standalone
Information technology	Security domains	Standalone
Information technology	Security mechanisms and countermeasures	Integrated
Information technology	Security services	Integrated
Information technology	Security vulnerabilities	Integrated
Information technology	Threat analysis	Integrated
Software engineering	Common vulnerabilities	Integrated
Software engineering	Cryptography	Standalone
Software engineering	Software risk and project management	Integrated

Table B6

Case 6 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	IA
Computer science	Ethics	Standalone
Computer science	Forensics	IA
Computer science	Incident handling	IA
Computer science	Network security	Integrated
Computer science	Privacy	IA
Computer science	Risk management	IA
Computer science	Security architecture	IA
Information technology	Attacks	Integrated
Information technology	Forensics	IA
Information technology	Fundamental aspects	Integrated
Information technology	Information states	Integrated
Information technology	Operational security	Integrated
Information technology	Policy creation and management	Integrated
Information technology	Security domains	Integrated
Information technology	Security mechanisms and countermeasures	Integrated
Information technology	Security services	IA
Information technology	Security vulnerabilities	Integrated
Information technology	Threat analysis	Integrated
Software engineering	Common vulnerabilities	Integrated
Software engineering	Cryptography	Integrated
Software engineering	Software risk and project management	Standalone

Table B7

Case 7 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	Integrated
Computer science	Ethics	IA
Computer science	Forensics	IA
Computer science	Incident handling	IA
Computer science	Network security	Standalone
Computer science	Privacy	IA
Computer science	Risk management	Integrated
Computer science	Security architecture	Standalone
Information technology	Attacks	Integrated
Information technology	Forensics	IA
Information technology	Fundamental aspects	Standalone
Information technology	Information states	Integrated
Information technology	Operational security	IA
Information technology	Policy creation and management	Standalone
Information technology	Security domains	Integrated
Information technology	Security mechanisms and countermeasures	Standalone
Information technology	Security services	IA
Information technology	Security vulnerabilities	Integrated
Information technology	Threat analysis	IA
Software engineering	Common vulnerabilities	Standalone
Software engineering	Cryptography	Standalone
Software engineering	Software risk and project management	Standalone

Table B8

Case 8 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	Standalone
Computer science	Ethics	Integrated
Computer science	Forensics	IA
Computer science	Incident handling	Standalone
Computer science	Network security	Standalone
Computer science	Privacy	IA
Computer science	Risk management	IA
Computer science	Security architecture	Integrated / Standalone
Information technology	Attacks	Integrated
Information technology	Forensics	IA
Information technology	Fundamental aspects	Standalone
Information technology	Information states	Integrated
Information technology	Operational security	Integrated
Information technology	Policy creation and management	Standalone
Information technology	Security domains	IA
Information technology	Security mechanisms and countermeasures	IA
Information technology	Security services	Standalone
Information technology	Security vulnerabilities	Integrated
Information technology	Threat analysis	IA
Software engineering	Common vulnerabilities	Integrated
Software engineering	Cryptography	IA
Software engineering	Software risk and project management	IA

Table B9

Case 9 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	IA
Computer science	Ethics	Integrated
Computer science	Forensics	IA
Computer science	Incident handling	IA
Computer science	Network security	Standalone
Computer science	Privacy	IA
Computer science	Risk management	IA
Computer science	Security architecture	IA
Information technology	Attacks	Standalone
Information technology	Forensics	IA
Information technology	Fundamental aspects	Standalone
Information technology	Information states	Standalone
Information technology	Operational security	Standalone
Information technology	Policy creation and management	IA
Information technology	Security domains	Standalone
Information technology	Security mechanisms and countermeasures	Standalone
Information technology	Security services	Standalone
Information technology	Security vulnerabilities	Standalone
Information technology	Threat analysis	Standalone
Software engineering	Common vulnerabilities	Integrated
Software engineering	Cryptography	Integrated
Software engineering	Software risk and project management	Integrated

Table B10

Case 10 Perceptions

Discipline	Topic	Perception
Computer science	Cryptography	Integrated / Standalone
Computer science	Ethics	Integrated
Computer science	Forensics	Integrated / Standalone
Computer science	Incident handling	Integrated / Standalone
Computer science	Network security	Standalone
Computer science	Privacy	Integrated / Standalone
Computer science	Risk management	Integrated / Standalone
Computer science	Security architecture	Integrated
Information technology	Attacks	Integrated / Standalone
Information technology	Forensics	Integrated
Information technology	Fundamental aspects	Integrated
Information technology	Information states	Integrated
Information technology	Operational security	Standalone
Information technology	Policy creation and management	Integrated
Information technology	Security domains	Integrated
Information technology	Security mechanisms and countermeasures	Standalone
Information technology	Security services	Standalone
Information technology	Security vulnerabilities	Integrated
Information technology	Threat analysis	Integrated
Software engineering	Common vulnerabilities	Integrated
Software engineering	Cryptography	Integrated
Software engineering	Software risk and project management	Integrated

Table B11

Description of participants

Participant	Title	Business Vertical	Education
Participant 1	Manager of Information Security	Medical Device Manufacturer	BS Electrical Engineering
Participant 2	Manager of Information Security	Third Party Logistics Brokerage	BS Organizational Management
Participant 3	Independent Security Consultant	N/A	BS Organizational Behavior
Participant 4	Director of Information Security and Compliance	Healthcare	BS Medical Technology
Participant 5	Managing Director	Security Consulting	BA Art History
Participant 6	Manager of Risk Advisories Services	Accounting and Auditing	High School / Military
Participant 7	Chief Information Security Officer	Specialty Healthcare	BS Business Management
Participant 8	Chief Technology Officer / Chief Security Officer	Computer Software	Masters Business Administration
Participant 9	Director of Information Security Assurance	Media and Information	MS Intructional Systems Technology
Participant 10	Independent Security Consultant / Instructor	N/A	General Education Diploma

Table B12

Inter-Discipline Analysis Per Case

Discipline	Topic	Number of Cases
Computer science	Cryptography	9
Computer science	Ethics	8
Computer science	Forensics	4
Computer science	Incident handling	5
Computer science	Network security	10
Computer science	Privacy	6
Computer science	Risk management	4
Computer science	Security architecture	7
Information technology	Attacks	11
Information technology	Forensics	n/a
Information technology	Fundamental aspects of security	10
Information technology	Information states	10
Information technology	Operational security	7
Information technology	Policy creation and management	8
Information technology	Security domains	9
Information technology	Security mechanisms and countermeasures	8
Information technology	Security services	7
Information technology	Security vulnerabilities	8
Information technology	Threat analysis	6
Software engineering	Common vulnerabilities	9
Software engineering	Cryptography	n/a
Software engineering	Software risk and project management	8
Information assurance	Cryptography	2
Information assurance	Ethics	2
Information assurance	Forensics	7
Information assurance	Incident handling	6
Information assurance	Network security	0
Information assurance	Privacy	5
Information assurance	Risk management	7
Information assurance	Security architecture	4
Information assurance	Attacks	0
Information assurance	Fundamental aspects of security	0
Information assurance	Information states	0
Information assurance	Operational security	3
Information assurance	Policy creation and management	2
Information assurance	Security domains	1
Information assurance	Security mechanisms and countermeasures	3
Information assurance	Security services	3
Information assurance	Security vulnerabilities	2
Information assurance	Threat analysis	4
Information assurance	Common vulnerabilities	1
Information assurance	Software risk and project management	2

Table B13

Intra-Discipline Analysis (Integrated and Standalone)

Discipline	Topic	Number of Cases (Integrated)	Number of Cases (Standalone)
Computer science	Cryptography	5	4
Computer science	Ethics	5	3
Computer science	Forensics	1	3
Computer science	Incident handling	3	2
Computer science	Network security	2	8
Computer science	Privacy	3	3
Computer science	Risk management	3	1
Computer science	Security architecture	3	4
Information technology	Attacks	7	4
Information technology	Forensics	1	2
Information technology	Fundamental aspects of security	5	5
Information technology	Information states	7	3
Information technology	Operational security	5	2
Information technology	Policy creation and management	2	6
Information technology	Security domains	4	5
Information technology	Security mechanisms and countermeasures	3	5
Information technology	Security services	1	6
Information technology	Security vulnerabilities	6	2
Information technology	Threat analysis	4	2
Software engineering	Common vulnerabilities	8	1
Software engineering	Cryptography	5	4
Software engineering	Software risk and project management	4	4

APPENDIX C. EXTERNAL DEFINITIONS

Attacks – Common attack vectors include the follow (NIST, 2012):

> -**External/Removable Media:** An attack executed from removable media or a peripheral device—for example, malicious code spreading onto a system from an infected USB flash drive.
>
> -**Attrition:** An attack that employs brute force methods to compromise, degrade, or destroy systems, networks, or services (e.g., a distributed denial of service intended to impair or deny access to a service or application; a brute force attack against an authentication mechanism, such as passwords, captchas, or digital signatures).
>
> -**Web:** An attack executed from a website or web-based application—for example, a cross-site scripting attack used to steal credentials or a redirect to a site that exploits a browser vulnerability and installs malware.
>
> -**Email:** An attack executed via an email message or attachment—for example, exploit code disguised as an attached document or a link to a malicious website in the body of an email message.
>
> -**Improper Usage:** Any incident resulting from violation of an organization's acceptable usage policies by an authorized user, excluding the above categories, for example; a user installs file sharing software, leading to the loss of sensitive data; or a user performs illegal activities on a system.
>
> -**Loss or Theft of Equipment:** The loss or theft of a computing device or media used by the organization, such as a laptop or smart phone.
>
> -**Other:** An attack that does not fit into any of the other categories. (NIST, 2012, p. 23)

Common Vulnerabilities - A weakness in a system, application, or network that is subject to exploitation or misuse (NIST, 2012, p. 50).

Cryptography – Cryptography is the study of symmetric and asymmetric encryption; digital signatures; cryptographic protocols, such as key exchange; cryptanalysis; and steganography (Pothamsetty, 2005; Lester, Narang, & Chen, 2008; Ferguson, Schneier, & Kohno, 2010).

Ethics - ethics as the following core topics: privacy issues (Cate, 2009); hacking and cracking; legal issues, such as security breaches and misuse; prevalent ethical dilemmas like whistle blowing; national or cultural differences; basis for ethical decision making; challenges in balancing freedom of information and security; security as a societal goal; and legal vs. ethical aspects (Cooper et al., 2010).

Forensics - This is typically differentiated from physical forensics taught in criminology by defining it as digital, computer, or information forensics. Digital forensics is the study of collecting, preserving, and reconstructing stored data evidence. This includes volatile data, nonvolatile data, and network packet captures from a computer or network where a suspected crime has occurred (Cooper et al., 2010).

Fundamental Aspects of Security – Defined by Machonachy, Schou, Ragsdale, and Welch (2001) as Security Services (Availability, Integrity, Authentication, Confidentiality and Non-repudiation), Security Countermeasures (Technology, Policy, and People), and Information states (Transmission, Storage, and Processing).

Incident Handling - The mitigation of violations of security policies and recommended practices (NIST, 2012, p. 12).

Information States – Information can either be transmitted, stored, or processed (Machonachy, Schou, Ragsdale, & Welch, 2001).

Network Security - The ACM and IEEE (2008b) define network security within the CS curriculum as the following: fundamentals of cryptography; authentication protocols; digital signatures; network attack types, such as denial of service, flooding, hijacking, etc; access control mechanisms; basic network defense tools and strategies, such as: intrusion detection, firewalls, Kerberos, IPSEC, virtual private networks (VPN), and network address translation (NAT); network management policies; and auditing and logging (Carlin & Gallegos, 2007).

Operational Security - event monitoring; access control; incident investigation; and policy enforcement (Baird & Gamble, 2010).

Policy Creation and Management – The ACM and IEEE (2008b) define policy creation and management as the following aspects: creation of policies; maintenance of policies; prevention; avoidance; incident response; and domain integration (physical, network, Internet, etc).

Privacy – The ACM and IEEE (2008b) defines privacy as topics around: HIPAA and FERPA, EU Data Protection, and Gramm-Leach-Bailey (GLBA).

Risk Management – According to Pironti (2008), Risk Management defines the areas of an organization's information infrastructure and identifies what information to protect and the degree of protection needed to align with the organization's tolerance for risk.

Security Architecture – Peterson (2006) defines Security Architecture as a unifying framework and reusable services that implement policy, standards, and risk management decisions.

Security Domains – The ACM and IEEE (2008b) defines security domains as the following aspects: human-computer interaction, information management, integrative programming, networking, program fundamentals, platform technologies, system administration, system integration / architecture, social issues, web systems, and physical plant.

Security Mechanisms and Countermeasures – Is defined by the ACM and IEEE (2008b) to include the following: cryptography, authentication, redundancy, and intrusion detection.

Security Services – Security services can be broken down into management (Security Program, Security Policy, Risk Management, Security Architecture, Certification / Accreditation, and Evaluation), operational (Contingency Planning, Incident Handling, Testing, and Training), and technical (Firewalls, Intrusion Detection, and PKI) (NIST, 2003).

Security Vulnerabilities - Security vulnerabilities are a weakness in a product that could allow an attacker to compromise the integrity, availability, or confidentiality of that product (Microsoft, n.d.)

Software Risk - Management of the various types of vulnerabilities. These problems can be exploited to cause considerable harm by external hackers or malicious insiders (Chess, 2002).

Threat Analysis - Threats are described as anything that would contribute to the tampering, destruction or interruption of any service or item of value. The analysis will look at every element of risk that could conceivably happen (Bayne, 2002)