https://doi.org/10.48009/3_iis_2016_116-123

Issues in Information Systems
Volume 17, Issue III, pp. 116-123, 2016

A MODEL SYSTEM AND SOFTWARE ASSURANCE GRADUATE CURRICULUM

Vladan Jovanovic, Georgia Southern University, vladan@georgiasouthern.edu
James Harris, Georgia Southern University, jkharris@georgiasouthern.edu
M. Tabatabaei, Georgia Southern University, jkharris@georgiasouthern.edu

ABSTRACT

A model graduate cyber security curriculum is presented that primarily designed to address the educational and training needs for developing secure software intensive systems. The curriculum is designed to satisfy the national requirements of the Information Assurance Workforce Improvement Program (per DoD8570.01-Mfrom 11.10.2015), for Tiers II and III jobs, primarily in the Systems Architecture and Engineering (IASAE) specialty areas. The curriculum maps to the knowledge bases of three approved Certifications for IASAE, namely the Certified Security Software Lifecycle Professional (CSSLP), the Certified Information Security Engineering Professional (ISSEP), and the Certified Information Systems Security Professional (CISSP). The course content is focused on four key elements: Security Architecture Analysis with Threat Modeling, Risk Assessment based on the Cyber Security Framework (and the NIST SP 800 series of guidelines), Secure Development Life Cycle under a process improvement framework CMMI, and a standardized set of Secure Coding best practices. This curriculum also addresses the making of formal Assurance Cases for Security and Safety of Software Intensive Systems, as well as Software Failure Mode Effect Analysis (FMEA).

Keywords: Curriculum, Cyber Security, Information Assurance

INTRODUCTION

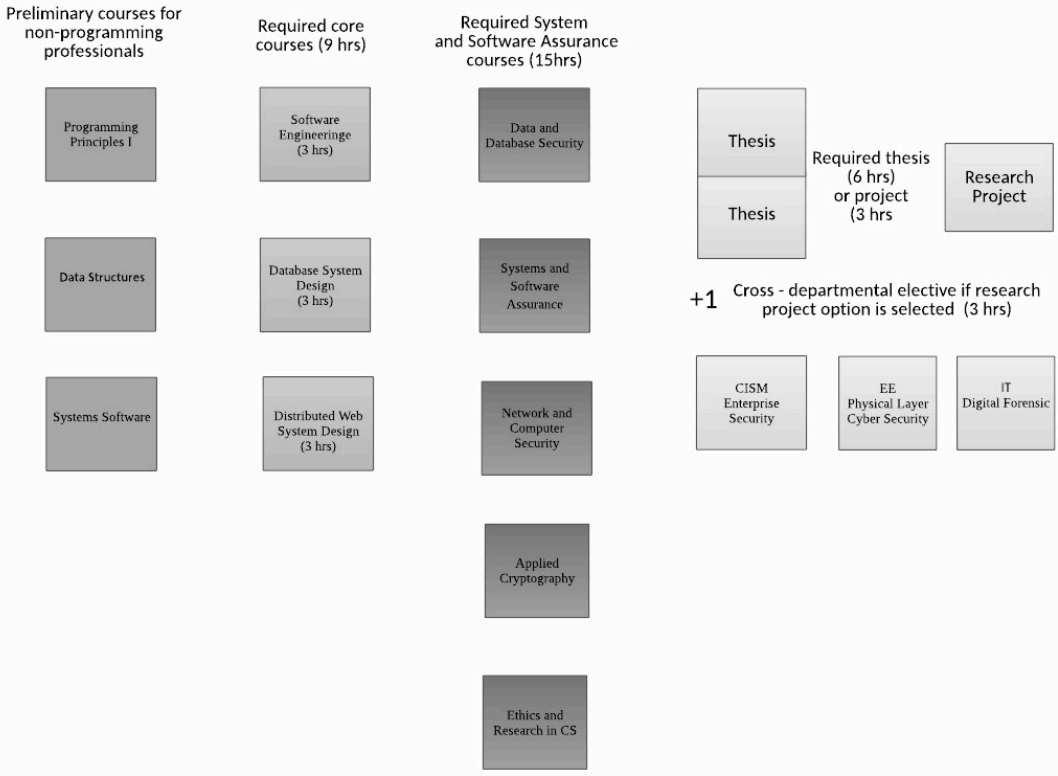
Due to demand, many new cyber security graduate programs are being started at colleges and universities. The focus of these programs is almost exclusively network security. There are few, if any, programs that focus on system and software assurance which is surprising considering the fact that all network and computer security related problems are inherently related to writing insecure code. The graduate curriculum presented in this paper is an online 30 hour program for professionals who wish to upgrade their credentials either by working towards a graduate degree in system and software assurance or by obtained the necessary knowledge to pass system and software assurance related certification exams. The program accommodates both programming and non-programming professionals by offering a set of preliminary courses that non-programming professionals must complete before being admitted. This has been successfully implemented in non-cyber-security related graduate CS fields attracting many capable professionals from technical fields where there is little or no programming, such as engineering. Non-programing professionals who wish to enter the program must complete three preliminary courses; a programming course, a data structures course, and a system software course. Each of these courses are programming intensive and are designed to develop programming skills. Students without programming aptitude are identified relatively quickly and are usually advised not to continue.

A DESCRIPTION OF THE PROGRAM

The main core of the program consists of three courses; Database System Design, Artificial Intelligence, and Distributed Web Systems Design. Each of these courses is three hours credit and are perquisites for the required system and software assurance curses (see Figure 1). These core courses ensure that graduate students who complete these courses are capable of programming at the graduate level, have a solid background in algorithms, and have been exposed to several different programming paradigms. The Database System Design course teaches a solid set of fundamentals including relational algebras, SQL, and normal forms. The Artificial Intelligence course is broken into two parts. Part I involves heuristic search algorithms, such as simulated annealing, A*, and genetic algorithms and part II covers knowledge representations such as frames, neural networks, and predicate logic. Heuristic algorithms and knowledge representations are becoming increasingly important in the movement of software systems from deterministic software systems to adaptable software systems. The Distributed Web Systems Design course teaches the fundamentals of a stateless (RESTful) Web client-server architecture. The course is usually taught using JavaScript as the client side language and PHP as the server side language. Stateless client-server Web architectures present a large set of unique software assurance problems not found in any other paradigms, foremost being the openness of the systems.

The required cyber security related courses include Applied Cryptography, Network and Computer Security, Data and Database Security, Ethics and Research in CS, as well as one additional elective from Computer Science, Computer Information Systems, Electrical Engineering or Information Technology.

Figure 1: Systems and Software Assurance Curriculum (30 hrs)



The Applied Cryptography course covers topics such as the fundamentals of cryptography, commonly used block and stream ciphers, hash functions, public key infrastructure, and current encryption standards. It is an applied course designed to make sure programmers can properly secure data through applying cryptographic techniques. The Network and Computer Security course is designed to give students the skills necessary to secure data and systems in a networked environment. The network security portion of the course follows up the TCPIP stack starting with physical security, the fundamentals of network architectures at the data-link layer (primarily Ethernet and 802.11 architectures) and their corresponding security issues, and TCPIP v.4 network protocols and their corresponding security issues TCPIP stack, including data-link layer protocols (Ethernet, 802.11), network layer protocols (ARP, IP), transport layer protocols (UDP, TCP), and application layer protocols (DHCP, DNS, HTTP). The computer security portion of the course covers security issues in commonly used operating systems (Windows, Linux), including mobile system (IOS, Android). The Data and Database Security course extends the Database Systems Design course to include how to secure data stored into and retrieved from databases. Topics include a review of relevant RDBMS security features (Oracle/SQL Server), protecting the models vs. data vs. applications, maintaining data integrity, access control (including TRBAC), SQL injections, privacy, auditing, and hardening and testing DB/DW Security (for both Oracle and SQL Server), and watermarking. The Ethics and Research in CS course is designed to make students aware of legal and ethical issues involved in creating secure software systems. The Data and Database Security, Applied Cryptography, and Network and Computer Security courses are prerequisites to the capstone System and Software Assurance course.

The capstone course, System and Software Assurance (SSA), is organized around the use of relevant standards/guidelines and organizational/system level processes and practices. The course is broken into two parts.

Part I deals with systems assurance and part II deals with software assurance (See Table I below for the course outline).

Table 1. Schedule of Topics and Assignments For System and Software Assurance Course

Module 1 Intro	**Orientation**
Week 1	- Scope, jobs, terms of reference, standards - Cyber Security Risk Management Framework *(Johnson, 2016), (NIST 800-37r1, 2010), (Conclin & Shoemaker, 2014), (Talabis & Marti, 2013), (Krutz & Vince, 2007)*
Module 2 System assurance	- System Assurance areas for software developers (including at organizational level), regarding Software/Data Intensive Systems
Weeks 2, 3, and 4: Security architecture - best practices in Threat Modeling	- System Architecture, Data Flow Models - Threat Modeling (TM): STRIDE, Threat Trees, CAPAC - Microsoft TM Tool 2016 laboratory practices - Elevation of Privilege Game - Analysis of architecture: six case studies - Linking TM with Risk, DRED. **HW1 Assignment: Threat Modeling-** 3 cases each team using MS TM tool. *(Shostak, 2014), (Schoenfield, 2015), (Berg, 2006), (Merkow & Raghavan, 2012), (Schumacher, Buglioni, Henderson, Buchman, & Somerland, 2006)*
Weeks 5, 6, and 7: Assessment methodology	- Regulation/Guidelines and Assessment Cases - Methods: IAM, RMF, ISO 27K,Octave, Fair - Assessing and improving a Security System - Defense in depth plan after gap analysis - Evaluation based on Common Vulnerabilities **Midterm Assignment- Assessment Project** *(Mills, 2004), (FIPS 199, 2004), (FIPS 200, 2008), (Johnson, 2016), (Talabis & Martin, 2013), (Anderson, 2008), (NIST 800-18, 2006) to (Nist 800-171, 2015)*
Week 8 Process Improvement	SDL Standardization, Process Capability Maturity Model CMMI, Assurance Maturity Model**,** Security Metrics: *(Shoemaker & Jovanovic, 2001), (Jovanovic & Shoemaker, 1999), (NIST 800-55, 2008), (NISTIR 8011, 2016)*
Week 9 System assurance review	**- Safety Assurance,** graduate students only - Safety/Hazard Assurance **HW2 Assignment- FMEA for Software** - Preparing and reviewing Assurance Cases:
Module 3 Secure coding	- Secure application coding in a networked environment, coding standards and common vulnerabilities.
Week 10 Designing security	- Principles: Compartmentalization, Reducing complexity, Redundancy, Auditing, and Coding standardization: Java, C/C++ *(Mohindra, Seacord, Sutherland & Svoboda, 2014), (McGraw, 2006) and numerous industry standards and guidelines (supported by MITRE, CERT, OSWAP, etc.)*
Week 11 Protecting access	- OS access control and least privilege - Validating input
Weeks 12, 13, and 14 Preventing common attacks	- Buffer, integer and float overflows - Error trapping (C, C++, C#, Java) - Cross-site scripting - SQL Injection - Other problems and remedies such as: race conditions, serialization, software integrity, sandbox, etc. plus using tools to detect flaws;
Week 15	- Security Code Review and Security Testing: *(Mohindra, Seacord, Sutherland & Svoboda, 2014), (McGraw, 2006), (Ransome, & Misra, 2014), (Merkow & Raghavan, 2012)*
Module 4 In parallel with Weeks 7-13	- Security related research in Software and/or Data Intensive Systems culminating a paper of publishable quality (following IEEE/ACM conference format) with a

	formal peer review.
Week 16	**Final Evaluation** Includes: term project presentation, secure coding project review, research paper peer review, and optional mock-up certification exam for selected relevant domains from the CISSP/CSSLP/ISSEP.

The SSA course requires both a high level of programming skill and a good knowledge of fundamental security issues and therefore requires the highest number of prerequisite courses in the program (core courses, Applied Cryptography, Data and Database Security, Network and Computer Security). The mappings of tables III, IV and V to security certificates shows the high relevancy of this course to the knowledge base of those certificates. For these reasons the course is designated as the capstone course within this program. It is illustrative to compare references from the Jovanovic (2008) version (designed to cover training requirements such as CNS 4014 (2004) and CNS 4016 (2006), and the current NIST guidelines (see references (NIST 800-18, 2006) to NIST 800-171, 2015)) as almost 70% of them are either new or have been substantially updated in the last eight years. The SSA course, in the context of our Computer Sciences (CS) curriculum (Jovanovic, 2013) has direct prerequisites (Ethics, Computer Security, Data Communication), as well as supporting coursework (Computer Architecture, System Software, Distributed Web Systems Design, Database Design, Object-Oriented Design, and Software Engineering capstone as required coursework). The program requires three courses out of following electives: SSA, Network Management, Software Testing and Quality Assurance, and Selected Topics relevant to security such as Storage/Cloud Technology courses derived from Jovanovic & Mirzoev (2012) and Jovanovic (2013).

RELEVANCE TO SYSTEM AND SOFTWARE ASSURANCE SPECIALIZATIONS

Since this program is designed for security professionals and those wishing to upgrade their credentials, the relevance of this program to obtaining security certification is illustrated. A number of certificates are required to reach performance levels required by many SSA jobs (see Table II). Approved certificates mapped to SSA jobs are shown for Information Assurance (IA), both Technical and Management (IAT and IAM), Computer Networks Defense Service Providers (CNDSP) specializations, and of primary interest, the IA System Architecture and Engineering (IASAE) job specializations.

The System and Software Assurance program prepares students for the Certified Information System Security Professional (CISSP) certification (see Table III). Of primary interest to future software developers is the Certified Secure Software Lifecycle Professional (CSSLP), a certification independent of the CISSP. Table IV lists mapping of CSSLP

to relevant portions of the System and Software Assurance program. In fact, the System and Software Assurance course alone maps very well to the desirable Information Systems Security Engineering professional (ISSEP) certificate (see Table V).

TABLE I. Job Categories vs. Selected Certification

Job/Level	Certificates
IAT Level I, IAT Level II	A subset of CISSP
IAM Level II, IAM Level III, IAT Level III, IASAE Level I	CISSP
IASAE Level II	CSSLP or CISSP
IASAE Level III	CISSP plus ISSEP/ISSAP
CNDSP: Manager; Analyst, Auditor, Infrastructure Support, Incident Responder	CISSP plus ISSPM; Certified Ethical Hacker (CEH)

TABLE II. Mapping CISSP to Relevant Coursework

CISSP 8- Domains	Relevant CS Course
Security and Risk Management	System & Sw. Assurance
Asset Security	System & Sw. Assurance
Security Engineering	Computer Security, Ethics

Communication and Network Security	Network and Computer Security, Storage Technology-Cloud Computing
Identity and Access Management	Network and Computer Security
Security Assessment and Testing	Systems and Software Assurance Data and Database Security
Security Operations	Network and Computer Security
Software Development Security	Systems and Software Assurance Data and Database Security Applied Cryptography Network and Computer Security

TABLE III. Mapping ISSEP To SSA Course

ISSEP 4- Domains	Systems and Software Assurance modules/assignments
Systems Security Engineering	M2 HW1- Threat Modeling, M2 HW2- Software FMEA M4 Research Paper, M3 Secure Coding Exercises, M3 Final Project, Plus - Network and Computer Security course, and elements from - Data and Database Security
Certification & Accreditation (Risk Management Framework)	M2 Midterm Project - Assessment Phase
US Government Information Assurance Related Policies and Issuances	M2 Midterm Project - Security System Gaps Analysis - Capability Improvement
Technical Management	M2 Midterm Project -Organization and Management Software Engineering capstone project course

TABLE IV. Mapping CSSLP To SSA Course

CSSLP 8- Domains	Systems and Software Assurance modules and/or supporting courses
Security Software Concept	M1 lectures/discussions
Secure Software Requirements	M2 HW1- evaluation only, Software Engineering course
Secure Software Design	M2 HW1- evaluation only, Data and Database Security
Secure Software Implementation/Coding	M3 Secure Coding Exercises M3 Final Project
Secure Software Testing	Software Testing & QA course – part of the Certificate
Software Acceptance	
Software Deployment, Operations, Maintenance and Disposal	Software Engineering capstone project course
Supply Chain and Software Acquisition	Software Engineering capstone project course

The System and Software Assurance program content and student outcomes are verified against relevant work specializations from the latest edition of the US NICE Workforce Framework (Workforce V2, 2015) that is replacing requirements from CNSS 4014 (2004) and CNSS 4016 (2005). Out of 31 work specializations of the Cyber Security Workforce (Workforce V2, 2015) the program covers over 13. A detailed mapping is under development and is expected to be presented at the conference and used in the process of applying for recognition as a center of excellence in cyber security education.

CONCLUSIONS

This paper presents an outline of a program in Systems and Software Assurance (SSA) together with extensive set of references that may be of use to others interested in similar security course for software system developers. It is our hope that educators may be interested in exchanging ideas, experiences, and relevant course materials. New detailed course outcomes, based on Jovanovic (2008) are in process of consolidation, and mapping to Workforce V2 (2015) as well as expectations per (Jovanovic, 2013), and will be available in October 2016. Most of the course content and material for the SSA course was redesigned, key of which was adding security architecture analysis with threat modeling and secure coding practices, and eliminating network evaluation as it is covered in the required Network and Computer Security course. The SSA course is being offered at the undergraduate level as a pilot, in the spring 2016. The initial reaction by a group of 20 undergraduate CS students is positive. Students appreciate the hands on model/code focus, both breadth and depth of the course, and especially the fact that course is being delivered by two experienced professors covering respectively systems and code level issues. The architectural analysis was considered the most elusive (in both variety of system types and multitude of security patterns (Shostak, 2014), (Schoenfield, 2015), (Berg, 2006), (Schumacher, Buglioni, Henderson, Buchman & Somerland, 2006) as the volume of material, especially NIST 800-39 (2011), NIST 800-53r4 (2013), NIST 800-53Ar4 (2014), NIST 800-64r2 (2008), and NISTIR 8011 (2016) was at times overwhelming. The recommended textbooks (Johnson, 2016), and (Long, Mohindra, Seacord, Sutherland, & Svoboda, 2014) were helpful and additional examples and guidance regarding in depth architectural analysis and threat modeling in (Shostak, 2014) and (Schoenfield,2015) are also quite effective. The graduate cyber security concentration within the Master of Science in CS, is approved and will share, as a core, the SSA course, include additional required coursework from CS (Applied Cryptography, Network and Computer Security, Data Systems Security, and the Ethics and Research in CS), and shared elective coursework with the CISM, IT and the EE departments (Enterprise Security, Cloud Security, Digital Forensic, and Physical Layer Security).

REFERENCES

Anderson R. (2008), *Security Engineering*, 2ed, John Wiley.

Berg C. (2006) . *High-Assurance Design- Architectures, Secure and Reliable Enterprise Applications.* Addison Wesley .

CNSS 4014 (2004). *National Information Assurance (IA) Training Standard for Information Systems Security Officers.*

CNSS 4016 (2005). *National Information Assurance (IA) Training Standard for Risk Analysts*.

Conclin, A., & Shoemaker D. (2014). *CSSLP Certification All-in-One Exam Guide*, McGraw Hill.

DoD 8570.01-M (2015). *Information Assurance Workforce Improvement Program*, updated 11.10.2015.

FIPS 199 (2004). *Standards for Security Categorization of Federal Information and Information Systems*.

FIPS 200 (2008). *Minimum Security Requirements for Federal Information and Information Systems*.

Johnson L. (2016). *Security Controls, Evaluation, Testing, and Assessment Handbook*, Elsevier.

Jovanovic V. (2013). Computer Science Curriculum 2013- An Example, *MIPRO 2013*, Opatia, Croatia, pp 759-764.

Jovanovic V. (2008).Systems Assurance Standards and Processes. *InfoSec Curriculum Development Conference 2008*, September 2008, Kennesaw, GA, pp 148-159.

Jovanovic V., & Mirzoev T. (2008). Teaching Network Storage Technology- Assessment Outcomes and Directions. *ACM SIGITE'08*, October 2008, Cincinnati, OH, pp 147-151.

Jovanovic V., & Mirzoev T. (2012). Teaching Storage Infrastructure Management and Security. *InfoSec Curriculum Development Conference*, October 2012, Kennesaw, GA, pp 41-43.

Jovanovic V.,& Shoemaker D. (1999). *ISO 9001 Standard and Software Quality Improvement, Benchmarking for Quality Management & Technology,* Vol. 4, No 2 1999, MCB University Press, pp 148-160.

Krutz R., & Vince R. (2007). *The CISSP and CAP Preparation Guide.* John Wiley.

Long F., Mohindra D., Seacord R., Sutherland D., & Svoboda D., (2014). *Java Coding Guidelines*. Addison Wesley.

McGraw G. (2006). *Software Security – Building Security In.* Addison Wesley.

Mansurov N., & Campara D. (2010). *System Assurance- Beyond Vulnerabilities.* MK 2010.
Merkow M., & Raghavan L. (2012). *Secure and Resilient Software.* CRC 2012

Mills G. (2004). Security Assessment Using NSA IAM. *Syngress 2004*.

Neufelder A. (2014). Effective Application of Software Failure Modes Effects Analysis. *SoftRel 2014*.

NIST 800-18 (2006). *Guide for Developing Security Plans for Federal Information Systems*.

NIST 800-27rA (2004), *Engineering Principles for Information Security (A Baseline for Achieving Security)*, Revision A, 2004.

NIST 800-30r1 (2012). *Guide for Conducting Risk Assessments*.

NIST 800-37r1 (2010). *Guide for Applying the Risk management Framework to Federal Information Systems*.

NIST 800-39 (2011). *Managing Information Security Risk*.

NIST 800-53r4 (2013) *Security and Privacy Controls for Federal Information Systems and Organizations*.

NIST 800-53Ar4 (2014). *Assessing Security and Privacy Controls in federal Information Systems and Organizations*.

NIST 800-55 (2008). *Performance Measurement Guide for Information Security*.

NIST 800-60 V1, r1. (2008). *Guide for Mapping Types of Information and Information Systems to Security Categories.*

NIST 800-64r2 (2008). *Security Considerations in the System Development Life Cycle.*

NIST 800-115 (2008). *Technical Guide to Information Security Testing and Assessment.*

NIST 800-137 (2011). *Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations*.

NIST 800-144 (2011). *Guidelines on Security and Privacy in Public Cloud Computing*.

NIST 800-160 (2014). *Systems Security Engineering*.

NIST 800-171 (2015). *Protecting Controlled Unclassified Information in Non-federal Information Systems and Organizations*.

NISTIR 8011 Volume 1 (2016). *Automation Support for Security Control Assessment*, Draft.

Ransome J., & Misra A. (2014). *Core Software Security.* CRC Press.

Schoenfield B. (2015). *Securing Systems- Applied Security Architecture and Threat Models.* CRC Press.

Schumacher M., Buglioni E., Henderson D., Buchman F., & Somerland P. (2006). *Security Patterns-Integrating Security and System Engineering.* J. Wiley.

Shoemaker D., & Jovanovic V. (2001) . *Engineering a better software organization*, Quest 2001.

Shostak A. (2014). *Threat Modeling: Design for Security*, J. Wiley.

Talabis M., & Martin J. (2013). *Information Security Risk Assessment Toolkit*, Elsevier.

Williams I., & Yuan X. (2015). Evaluating Effectiveness of Microsoft Threat Modeling Tool, *ISCD Conference 2015*, Kennesaw GA, October 2015.

WorkForce V2. (2015). retrieved February 7. 2016 from the: https://niccs.us-cert.gov/training/tc/framework/specialty-ares.