Proceedings of the 12th Colloquium for Information Systems Security Education
University of Texas, Dallas
Dallas, TX June 2 - 4, 2008

Designing a Security Thread in Computing Curricula

Ju An Wang, Max North, *Southern Polytechnic State University*, and Sarah North, *Clark Atlanta University*

Abstract – Information security is one of the pervasive themes in computing curriculum. As computing security becomes more important in all sectors of society, so does the preparation of our students with knowledge and understanding of critical security concepts, methodologies, and techniques. Unfortunately, despite the deep and pervasive impact of security, undergraduate computing curricula and programs today often look much as it did several decades ago. We want to infuse information security into our computing curriculum, and we found a good model for doing that. This paper introduces the Threads model for computing curriculum originated from Georgia Tech's College of Computing, an innovative way to restructuring computing curriculum. We believe that a security Thread should be developed for any undergraduate computing programs. We discuss the rationales, design, and implementation for an information security thread in computer science, software engineering, and information technology programs as well as the challenging issues we have faced.

Index terms – Information security, Computing curriculum, Threads model, Information security education.

I. Introduction

Undergraduate enrollments and degree production in computing have shown a dramatic up-and-down pattern over the past 30 years. According to a recent study conducted by the Computing Research Association (CRA), the number of newly-declared computer science majors in the fall of 2007 was half of what it was in the fall of 2000 – 7,915 versus 15,958. Enrollments in computer science programs have fallen for several years as well. However, the IT industry continuously grapples with a worldwide skills shortage that does not look like it will get any better any time soon as the number of people graduating with computer science degrees is about to hit an all time low. Skills relating to security, firewalls, and data privacy are most in demand according to a recent survey done by CompTIA entitled “Worldwide State of IT Skills” [1]. Information security is one of the cornerstones of the information society. Security is seen as the most desirous IT skill worldwide, and many companies find their in-house security skills lacking [2, 3, 4]. The CompTIA study surveyed more than 3,500 IT professionals from around the world, and found that a large portion of U.S. companies want more highly skilled workers, especially in the areas of security, networking and operating systems.

Computing has become an enabling technology for other sciences and technologies. As secure computing becomes more important in all sectors of society, so does the preparation of our students with knowledge and understanding of critical security concepts, methodologies, and techniques. Unfortunately, as [11] points out, despite the deep and pervasive impact of security and other new technologies, undergraduate computing curricula and programs today often look much as it did several decades ago. We have to revitalize our undergraduate computing education to generate a U.S. workforce with the computing competencies and skills imperative to the Nation's health, security and prosperity in the 21st century. It has become increasingly clear that old computing curricula were not effective in generating graduates who remain competitive in an increasingly global, interconnected economy. We must revise and re-design our computing curricula to address the common challenges such as fluctuating enrollments in traditional computer science programs, changes and trends in workforce demographics, the imperative to integrate fast-paced computing innovations into curriculum, and the need of a diverse, agile workforce with the computing security knowledge essential to U.S. leadership in the information warfare and global innovation enterprise.

At the same time, there have been several studies exploring how computer science is perceived by underrepresented groups [13, 14]. With the outsourcing and off-shoring continuously shuffle programming jobs out of U.S., students report that they do not see the relevance of what they are studying and they are looking for innovative curriculum to arm them with knowledge to compete and succeed in the increasingly competitive global economy. It has become clear that the old way of doing things is not going to work. Students graduate from old computing programs will find it difficult to differentiate themselves and remain competitive in an increasingly dynamic economy. In a pioneering effort, The College of Computing at Georgia Tech has been working over the past several years to develop a paradigm-changing undergraduate computing curriculum model called Threads, which promises to empower students to continuously add value throughout their careers, be adaptive to changing information technology value propositions, avoid being outsourced, and possess a global mindset [6, 7, 15].

The Threads model represents a natural evolution of contextualized computing education. It includes an infrastructure for intentional advising, as well as the development of robust software support for administrators, advisors, educators, and students. Threads, as a curriculum model, represents both a process for understanding and developing curricula, and a set of outcomes derived from the application of the process [15].

With the support of NSF CPATH grant, we started to evaluate and adapt the Threads model in our computing curriculum at School of Computing and Software Engineering, Southern Polytechnic State University. The major goals of this project include: (1) to assess the application of Threads and its supporting programs developed by Georgia Tech; (2) To study the application of the Threads model to a diverse computing programs including Computer Science, Software Engineering, and Information Technology; and (3) to disseminate the results, experiences assessment tools, software support infrastructure, and development process to the larger computing community. With our initial study, found the Threads model is a great tool to revitalize computing curriculum. We are introducing a security thread in our computing curriculum along with a couple of innovative threads eDvice and Intelligent Computing. This paper will focus on the Security thread.

The rest of the paper is organized as follows: Section II gives a brief introduction to the Threads model and the current eight threads implemented in Georgia Tech. Section III discuss the process of designing, implementing, and assessing the Threads model in Southern Polytechnic State University and Clark Atlanta University. Section IV proposes a new Security thread, and finally, we discuss some of the issues we are facing and further research topics in the last section.

II. What is The Thread Model?

The Threads model was developed in the College of Computing at Georgia Tech as a new approach to undergraduate computing education in the face of an increasingly competitive global environment [6]. Threads promises to empower students to define their computing identities, take ownership of their career trajectories, and pursue enhanced international and diversity experiences. Under the Threads curriculum model, computing students will develop expertise in multiple spheres; recognize and synthesize patterns; cross boundaries; be able to corral diverse technologies, disciplines, individuals, and organizations and produce a unified, value-added solution; and make bold leaps toward innovating and inventing something new from previously uncombined elements [6, 7].

The major purpose of the Threads model is to increase the value of an undergraduate computer science degree, to produce graduates who will be in high demand and who will continuously contribute value throughout successful careers. Threads graduates have a broad set of skills and are not easily outsourceable.

The Threads model has two components: (1) students' computing identities, which are defined by two intertwined pathways through the program, and (2) students' computing trajectories, which represent what they want to become or how they want to apply their knowledge in the real world. Threads embody a set of broad, horizontal skills that live within and outside of computing. Currently there are totally eight Threads as described below.

(1) **Devices** – where computing lives in the world. The devices thread is concerned with embedded computational artifacts that interact with people or the physical world. In this thread one learns how to create and evaluate devices that operate under physical constraints such as size, power and bandwidth. Examples include PDAs, cell phones, robots, jet engines, and intelligent appliances. Students learn how to build computing devices, what computing devices to build, and how to use computing devices once they have been deployed. After completing the devices thread, there will be no more mystery left in the computer for the student.

(2) **Information Internetworking** – where computing meets data. This thread deals with computing and data with its implications in the context of personal and organizational information management. This thread prepares students for all levels of information management by helping them to capture, represent, organize, transform, communicate, and present data so that it becomes information, using geographically distributed multi-national organizations and their information needs as context.

(3) **Intelligence** – where computing models intelligence. This thread is concerned with top-to-bottom computational models of intelligence. To this end, it emphasizes designing and implementing artifacts that exhibit various levels of intelligence as well as understanding and modeling natural cognitive agents such as humans, ants, or bees. Students acquire the technical knowledge and skills necessary for expressing, specifying, understanding, creating, and exploiting computational models that represent cognitive processes. The thread prepares students for fields as diverse as

artificial intelligence, machine learning, perception, and cognitive science, as well as for fields that benefit from applications of techniques from those fields.

(4) **Media** – where computing meets design. The media thread prepares students by helping them to understand the technical and computational capabilities of systems in order to exploit their abilities to provide creative outlets.

(5) **Modeling and Simulation** – where computing models the world. This thread is intended for students interested in developing a deep understanding and appreciation of how to create simulations and other computational models.

(6) **People** – where computing meets users. The People thread prepares students by helping them to understand the theoretical and computational foundations for designing, building, and evaluating systems that treat the human as a central component.

(7) **Platforms** – where computing builds itself. The platforms thread is where many of the practical skills of computing are learned, such as computer architecture and organization, how they are built using different hardware and software layers and how they are programmed.

(8) **Theory** – where computing models scaling. The theory thread is where computing models and addresses scaling. Theory quantifies, in mathematical terms, the efficiency by which problems are solved, as problem instances grow in size.

Any two Threads can be intertwined, leading to a degree in computing, as presented in the following formula:

Thread_1 × Thread_2 → B.S. Degree in Computing

Threads combine regular computer science instruction with additional classes related to a particular area of application, targeting a real-world computing opportunities for a computing graduate. Threads offers students the opportunity to pursue and prepare themselves for a great variety of computing areas, such as gaming and entertainment computing, intelligent robotics, multimedia and animated movies, mobile computing, bioinformatics, and so on. In the original Threads model description [6], four Roles were defined: Master Practitioner, Entrepreneur, Innovator, and Communicator.

A thread serves as a context for interpreting the courses in a curriculum for both students and faculty. A thread makes its set of courses cohesive and suggests a coordinated path through its courses so that the end result is expertise in the area of the thread. Each thread is about two-third of a degree, but any pair of threads yields a complete degree.

III. The Designing Process

Our project is to adapt, extend, and evaluate Threads as a computing curriculum model in our universities. We would like to use Threads as a useful model of curricular reform to improve enrollment and retention and to broaden the perceived vision of computing. Very often, undergraduates find making course choices intimidating. We expect the new Threads model to support and enhance mechanisms for advising students.

Southern Polytechnic State University (SPSU) is a four-year public residential institution within the University System of Georgia. Approximately 4,200 traditional and non-traditional students are enrolled for day, evening, and weekend classes pursuing bachelors and master’s degrees as well as academic credit certificate programs in a variety of technology, math and science-related fields. In 2006 SPSU was ranked #1 nationwide in the number of bachelor’s degrees in engineering technology awarded to African American students, #4 in the number of students enrolled in engineering technology programs and #5 in the number of engineering technology degrees awarded to women by “Profiles of Engineering and Engineering Technology Colleges”, published by the American Society for Engineering Education. Based on our student population, we would like to investigate the impact of Threads on non-traditional students, and how such a Threads model will affect our connections with DTAE (Department of Technical and Adult Education) schools in Georgia and their curricula.

We identified four phases in a process of designing and implementing a Threads-based curriculum.

1. Design Phase

This phase includes works like talking to students, faculty, target industry and other program constituents about Threads model and briefing administrators about Threads model to seek their support. After the conceptual work of possible threads is completed, the necessary documentation will be filed for the curriculum committee to approve.

2. Implementation Phase

The phase involves formal course and curriculum development. A process and procedure have to be in place for updating and modifying threads as needed. The Threads curriculum has to merge with the academic planning process or strategic planning process. It must be consistent with the long-term schedule and required resources must be obtained to implement it.

3. Advising Phase

Advising is one of the most important services faculty can provide to students to make smooth transition from traditional curriculum to a Threads-based curriculum. Our project aims at a systematic way to advise our students in the development of meaningful educational plans and achievement of personal and educational goals. We will use software tools, e.g., Threadspace, and develop necessary advising infrastructure to help students benefit from the new Threads curriculum model.

4. Assessment Phase

This phase includes the development of appropriate evaluation instruments and a process for carrying out such evaluation and assessment and feedback mechanisms to improve the Thread model continuously. We will conduct formal evaluation about our project in a number of areas including academic performance, intellectual growth, student engagement, enrollment, retention, and whether Threads better prepare students for their future careers.

IV. The Security Thread

The security thread is where computing meets the requirements of confidentiality, integrity, and availability.

The security thread is concerned about how to build trustworthy computing systems, and how to protect information systems from malicious attacks, such as interception, fabrication, modification, and interruption. Information security rests on confidentiality, integrity, and availability. In this thread, students will learn how to create information systems satisfying security requirements, and how to effectively defend computing systems by various approaches and techniques. Building secure systems has to evolve from an art to a security engineering discipline, with well-defined methods for constructing secure systems out of secure subsystems and basic components, and for assessing and formally validating security. Security-preserving notions of composability need to be developed, combining techniques from software engineering, secure hardware design, formal methods and cryptography.

Students who pursue security can combine it with eDevice thread (a new thread under development) to study secure hardware design, or software engineering to build trustworthy software systems. Examples of positions for students graduated with security thread include penetration testers, security engineers, CSO (Chief Security Officer), CISO (Chief Information Security Officer), chief security managers, security operations managers, directors of security, fraud investigators, security managers, security administrators, etc.

The early preparation for the security thread includes the following:

- Programming and design
- Linear algebra
- Discrete math
- Computer architecture and organization
- Computer networking
- Databases
- Platforms and web technology

Thread Learning Objectives

The thread learning objectives are statements that describe the career and professional accomplishments that the thread is preparing graduates to achieve, focusing on competencies of graduates. The security thread will enable its graduates to

(1) Be effective in the design of security solutions and the practical applications of security principles;
(2) Exhibit fundamental skills in programming, platforms, networking and data management;
(3) Be characterized by effective leadership skills and high standards of ethics;
(4) Be successfully employed or accepted into graduate programs;
(5) Engage in life-long learning and serve their communities.

Thread Outcomes

The thread outcomes are narrower statements that describe what students are expected to know and be able to do by the time of graduation. The four thread outcomes for the Security Thread are listed below, which are minimal set of skills for our students to have:

(1) Ability to identify and analyze security issues in information systems;
(2) Ability to evaluate the impact of business constraints and processes on the implementation of Information Security solutions;
(3) Ability to design and implement security solutions;
(4) Ability to address a specialized area of information security management.

Required Courses (39 hours)

- MATH 2260 Prob & Stat
- MATH 2345 Discrete Math
- CSE 1301 Computer Science I
- CSE 1302 Computer Science II or IT 1324 Advance Programming Principles
- IT 3423 Operating Systems
- CSE 4813 Introduction to Computer Security
- IT 4823 Information Security Administration
- IT 4833 Wireless Security
- IT 4843 Ethical Hacking and Network Security
- IT 4853 Computer Forensics

- IT 4863 Database Security
- IT 4873 Information Security Design and Implementation
- IT 4903 Special Topics in Information Security

Pick 2 from the following (6 hours)
- CS 3153 Database Systems
- IT 3123 Hardware and Software Concepts
- IT 4323 Data Communications and Networks
- IT 3203 Introduction to Web Development

Pick 2 from the following (6 hours)
- IT 4203 Advanced Web Development
- IT 4153 Advanced Database
- IT 4333 Network Configuration and Administration
- IT 3653 Client Server System Administration

Pick 1 from the following (3 hours)
- IT 4723 IT Policy and Law
- IT 4683 Management Information Systems
- IT 3223 Software Acquisition and Project Management

Free Electives (6 hours)
- Any courses listed above but not picked
- IT 4123 Electronic Commerce
- SWE 2642 Professional Practices and Ethics
- SWE 4324 User Centered Design
- SWE 4663 Software Project Management
- SWE 4724 Software Engineering Project
- IT 4903 Special Topics in Information Technology

V. Conclusion and Discussion

Computing has permeated and in many cases transformed almost all aspects of our everyday lives. Trustworthy computing is essential to protect our national infrastructures, homeland security and prosperity in the 21st century. We believe that lack of context may explain some of the decline in student interests in computing. Georgia Tech's Threads computing curriculum model represents a new structuring principle and a natural evolution of contextualized computing education. It is necessary to design and implement a security thread under the overall Threads model for a computing school to produce graduates who will be in high demand and who will continuously contribute value throughout successful careers. The graduates from a Threads-based curriculum should have a broad set of skills and they are not easily outsourceable.

Several issues have to be considered when we introduce a security thread to our computing programs, and adopt Threads model in general. Below we briefly discuss some of the challenges we are facing with some of our suggested solutions.

1. Thread and Mandatory Advising

The School of Computing and Software Engineering at Southern Polytechnic State University recognizes the crucial role of advising in ensuring that students progress towards successful graduation. Advising supports and facilitates the student success through the appropriate sequencing of courses, and knowledge of available career opportunities and an awareness of the skills and knowledge required to take advantage of these career opportunities. To this end, the School developed a mandatory advising policy that requires each student to have a degree completion plan, which specifies that in which semester the student will complete a particular course, and a student must seek advising to create a degree completion plan covering the next two calendar years. In order to make our Threads curriculum work with our mandatory advising policy, we need an automated tool to help us to manage the degree, integrating changes while maintaining consistency and ensuring accreditation. Currently we are looking at DegreeWorks product and the software tool, ThreadSpace, developed at Georgia Tech for this purpose.

2. Threads and Administrative Support

As described in [6], Threads represents a tremendous departure from how people think about curricula in computing, a departure from a vertically-oriented curriculum whose goal is the creation of students with a fixed set of skills and knowledge. The Threads model does not following the old way of monolithic core curriculum plus a number of elective courses for students to choose. This significant change brings in many academic as well as administrative challenges and consequences. Georgia Tech College of Computing has 68 academic faculty and 39 research faculty. The college has over 1,200 undergraduates, 200 Master's students, and 300 Ph.D. students. Needless to say, they have a big budget and they can afford the luxury of offering eight threads with flexibility. For a small school or even a small department, however, it will be challenging to offer a curriculum allowing students to select freely any two threads out of eight while the number of faculty and resources are limited. Therefore, it is important to maintain an appropriate size of the Threads and secure strong support from the administration for adequate resources to make the Threads model successful.

3. Threads and Accreditations

Many computing programs are seeking accreditation or reaffirming their accreditation to ensure that education provided by them meets acceptable levels of quality. Under the Threads model, the combination of any two threads comes up with a degree. This means that we have to design our threads in such a way that the union of any two threads must satisfy the accreditation requirements for a computing degree. In terms of thread outcomes, this means that we have to design our threads such that all general program outcomes (a) – (i) listed in [21] must be satisfied as well as program specific outcomes for individual computing programs. This is challenging but doable according to [22]. As suggested by one of the reviewers, a further research question is to look at the mapping between Threads and the coverage of the ACM recommended computing curricula [4].

4. Transfer and Compatibility

As a further research topic, we must find out exactly how different is the Threads model compared with more conventional or traditional programs in computing, such as Computer Science, Software, Engineering, Information Systems, Information Technology, etc. In particular, will Threads allow students to transfer to other universities without Threads? Will Threads allow students to transfer to other departments with their own possibly different threads?

At SPSU, we have a strong relationship with our feeder schools including community colleges and Georgia DTAE (Department of Technical and Adult Education, 2-year colleges offering Associate degrees) schools. Is it possible to involve such 2-year colleges in a Threads model? We should design our Threads model in such a way so that students could transfer from a community or technical college that does not follow the Threads program and still engage in Threads for their last two years. We would like to investigate a multi-staged Threads model that retains the current pipelining structure between our schools and the DTAE schools.

5. Threads and Concentrations

Threads are not tracks or concentrations. Two concentrations always share the same core while two threads do not have an intersected core. Every student constructs her own personalized computing degree by weaving two threads. While two concentrations will not generate a degree, two threads, however, together will fulfill the requirements for an accredited degree. In describing a computing degree using Threads, one can avoid asking what courses must every computing student take, instead, we seek to identify a set of threads that make sense to students and to the employers and graduate schools, while maintain the emphases and strengths of our school or department. This could be a daunting task.

Another interesting question is: What is the relationship between a thread and a minor? If a non-computing student wants to have a minor in computing (either CS, IT, SWE, or IS), how does she fit into this Threads curriculum?

6. Unifying IT and CS

Information Technology (IT) is a sub-area in the computing discipline with an emphasis on technology. As an academic discipline, IT focuses on meeting the needs of users within an organizational and social context through the selection, creation, application, integration, and administration of computing technologies [4]. The set of pervasive themes in IT are similar to those found in computer science – such as information security and assurance, user centeredness and advocacy, communication and leadership skills – but differ in some important ways. Can a Threads model provide a unifying framework for IT and CS education? Would such a framework combine the two areas? Is there a way to create Threads just within an IT degree? There are several possible thrusts within IT that might support a threaded IT program: healthcare, biomedical information systems, e-government, digital forensics, entertainment computing, retail-chains, and enterprise resource management. It is worthwhile to investigate how to unify computing and non-computing as well.

VI. Acknowledgments

The research work was supported by the NSF CPATH grant #0722157.

The authors would like to acknowledge all the three anonymous reviewers of this paper for their constructive comments and valuable suggestions on an earlier draft of this paper.

VII. References

[1] Karen D. Schwartz, Security Tops List of Most Wanted, Most Lacking IT Skills, eWeek, 2008-2-27, www.midmarket.eweek.com.
[2] Charles W. Reynolds and Bryan S. Goda, The Affective Dimension of Pervasive Themes in the Information Technology Curriculum, Proceedings of SIGITE'07, October 18-20, 2007, Destin, Florida, USA. ACM 978-1-59593-920-3/07/0010, pp. 13 – 20.
[3] ACM and IEEE Computer Society . Computing Curricula 2005. The Overview Report. ACM and IEEE Computer Society, 2005. (at http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf)
[4] ACM Computing Curricula, Information Technology Volume. ACM, 2005. (at

http://www.acm.org/education/curric_vols/IT_October_2005.pdf)

[5] Meyer, Jan and Ray Land. Threshold Concepts and Troublesome Knowledge: Linkages to Ways of Thinking and Practising within the Disciplines. Occasional Report 4, ETL Project, Universities of Edinburgh, Coventry and Durham, May 2003.

[6] Merrick Furst and Richard DeMillo, Creating Symphonic-Thinking Computer Science Graduates for an Increasingly Competitive Global Environment, White Paper, Georgia Institute of Technology, College of Computing, http://www.cc.gatech.edu/education/undergrad/bscs/threads_whitepaper.pdf.

[7] Threads Resources, College of Computing, Georgia Institute of Technology, www.cc.gatech.edu/education/undergrad/bscs, accessed on March 2, 2008.

[8] Swiderski, F. and Snyder, W., *Threat Modeling*, Microsoft Press, 2004.

[9] Mullins, P. et al "Panel on Integrating Security Concepts into Existing Computer Courses", SIGCSE '02, Feb. 27 - Mar. 3, 2002, Covington, KY, 2002.

[10] Davidson, M.A. "Leading by Example: the case for IT Security in Academia", Educause Review, Jan/Feb. 2005.

[11] NSF Program Solicitation, CISE Pathways to Revitalized Undergraduate Computing Education (CPATH), NSF 06-608, http://www.nsf.gov/pubs/2006/nsf06608/nsf06608.htm.

[12] Wang, J.A., “Security Testing in Software Engineering Courses”, *Proceedings of Frontiers in Education Conference*, Session F1C, IEEE Catalog Number 04CH37579C, ISBN: 0-7803-8553-5. October 2004.

[13] AAUW, Tech-Savvy: Educating Girls in the New Computer Age, American Association of University Women Education Foundation, New York, 2000.

[14] J. Margolis and A. Fisher, Unlocking the Clubhouse: Women in Computing, MIT Press, Cambridge, MA, 2002.

[15] M. Furst, C. Isbell, and M. Guzdial, Threads: How to Restructure a Computer Science Curriculum for a Flat World, in Proceedings of the Thirty-Eighth Technical Symposium on Computer Science Education (SIGCSE), 2007.

[16] NSTISSI-4011, National Training Standard for Information Systems Security (INFOSEC) Professionals, dated 20 June 1994. http://www.cnss.gov/instructions.html.

[17] CNSSI-4012, National Information Assurance Training Standard for Senior systems Managers, dated June 2004, http://www.cnss.gov/instructions.html.

[18] CNSSI-4013, National Information Assurance Training Standard for System Administrators (SA), dated March 2004. http://www.cnss.gov/instructions.html.

[19] CNSSI-4014, National Information Assurance Training Standard for Information Systems Security Officers, dated April 2004. http://www.cnss.gov/instructions.html.

[20] CNSSI-4016, National Information Assurance Training Standard for Risk Analysts, dated November 2005. http://www.cnss.gov/instructions.html.

[21] ABET, 2008-2009 Criteria for Accrediting Computing Programs – New Criteria, 2008. http://www.abet.org/.

[22] Rich LaBlanc, Threads Assessment and ABET Accreditation, presented at the First Workshop of CPATH, Georgia Tech, Atlanta, January 24, 2008.