Empowering Faculty to Embed Security Topics into Computer Science Courses

Ambareen Siraj, Sheikh Ghafoor, Joshua Tower, Ada Haynes
Tennessee Technological University
Department of Computer Science
110 University Drive
Cookeville, TN 38505
{ASiraj, SGhafoor, JDtower21, AHaynes}@tntech.edu

ABSTRACT

Security illiteracy is a very common problem among Computer Science (CS) graduates entering the nation's digital workforce, which has contributed to a national cyber-infrastructure that could and should be more resilient to cyber-enemies than it is now. The Security Knitting Kit (SecKnitKit) project aims to improve security awareness, knowledge, and interest of undergraduate CS students by exposing them to computer security concepts and issues in their regular course of study. The project is developing, deploying, and disseminating a multi-faceted out-of-the-box instructional support system to empower non-security faculty. These are faculty who have no experience in teaching security but recognize the importance of security in today's world and want to broaden their teaching repertoire. This project enables them to weave relevant security topics traditional computer science courses seamlessly and effectively. The project is organized by the CS department at Tennessee Tech University (TTU) and supported by the National Science Foundation under grant DUE-1140864.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education

General Terms

Security, Experimentation

Keywords

Security, Security Education, ACM/IEEE-CS 2013 Curricula

1. INTRODUCTION

"*At least some computer security instruction should be a prerequisite in participating in the information age*" – Dr. C.Y. Irvine, Challenges in Computer Security Education," IEEE Software, Sept./Oct. 1997

Lack of information assurance (IA) and security in deployed computer software and in daily operational usage of software costs businesses and taxpayers severely every year. The CSI Computer Crime and Security Survey [3] reports that the average loss by an organization in 2009 was close to a quarter million dollars; and 60% of these financial losses were attributed to non-malicious actions by "insiders." These insiders are mostly end users and computer professionals with operational and developmental responsibilities. Examples include software engineers that leave exploitation opportunities for buffer overflow without proper input validation; application developers who do not implement access controls; and system users who do not ensure appropriate security configurations. Although it has become critical that "all" Computer Science (CS) students need to experience security training in traditional subject matters, at the least as threaded security topics, most higher education institutions do not have the faculty capacity to teach specialized security courses. The majority of institutions offering security courses at the undergraduate level offer them as standalone individual courses. In other words, security is treated in isolation. Since not mandatory in CS curriculum, many CS undergraduates can successfully achieve their degree without being exposed to any security courses during their course of study. Only 129 out of more than 4,000 higher education institutions in the United States are accredited by the government as National Centers of Academic Excellence in Information Assurance Education Programs (CAE/IAE) to specialize in computer security education [9]. With the absence of required security courses in traditional CS curriculum, the vast majority of graduates enter the digital workforce with no knowledge or basic understanding of information security – one of the essential skill sets for the 21st century. In addressing this concern, Information Assurance and Security (IAS) has been designated as a new knowledge area in the new ACM/IEEE-CS Curricula 2013 [1]. The guideline denotes that IAS "is added to the Body of Knowledge in recognition of the world's reliance on information technology and its critical role in computer science education". Also, for an academic institution to be accredited as CAE/IAE, one of the criteria that needs to be fulfilled is "Criteria 2: IA Treated as a Multidisciplinary Science" [8], which clearly states that the academic program under consideration must demonstrate that IA is not treated as a separate entity but is incorporated into existing courses and non-IA students are introduced to IA concepts. Since security concepts are not integrated into existing curriculum in the majority of institutions - the aspiration of accreditation remains beyond their reach. This leads to another key motivation behind this project, which is providing support for interested educational institutions to move closer towards CAE/IAE accreditation.

Since in today's world security consciousness has become an essential skill set and part of good citizenship [7], the primary goals of this project are:

- To improve security awareness, knowledge and interest of undergraduate computer science students by exposing them to computer security concepts and issues in their regular course of study ***(student learning goal)***.
- To improve the security awareness and security teaching expertise for non-security faculty ***(faculty expertise development goal)***.

ITICSE'14, June 21–25, 2014, Uppsala, Sweden.
Copyright © 2014 ACM 978-1-4503-2833-3/14/06…$15.00.
http://dx.doi.org/10.1145/2591708.2591741

- To promote the use of security integration strategy and materials in other institutions ***(dissemination goal)***.

Security Knitting Kit (SecKnitKit): Integrating Security into Traditional Computer Science Courses" is a National Science Foundation supported project (DUE Award#-99999) that develops, deploys and disseminates a multi-faceted out-of-the-box instructional support system to empower non-security faculty (faculty whose primary teaching/research focus is not security) in CS to seamlessly and effectively weave security topics into traditional CS courses. The project is implemented by the CS department at Tennessee Tech University (TTU).

2. Related Work in Integrating Security into Traditional Courses

Primarily, there are three models of security education [11]: a security track; a single security course; and threading security topics into traditional CS courses.

Security specialization with a security track is used in the graduate curriculum and is intended to build and train cyber-warriors. The single security course approach is typically offered as an elective in the undergraduate curriculum. Not all institutions offer such dedicated courses because faculty expertise in this area is not always available and adding an extra security course to a full curriculum is very difficult. The unconventional third model, which addresses the problem of integrating security into CS curriculum, is the focus of this project.

Integrating security topics in the traditional CS curriculum is a well-recognized, challenging problem that has been the focus of many scholarly works [4,7,10,11,16-18]. Scholars have articulated this need, and in some cases shared ideas, observations, and recommendations, however, very few actual initiatives have been undertaken or implemented to solve this problem.

Graduate students at the University of Maryland, Baltimore County conducted a project where they proposed to create security modules for programming classes like CSI, CSII and data structures [12]. In Georgia State University, students are taught to consider information security through real world scenarios and relevant exercises within CSI [6].

Funded by NSF grant DUE-0817267, "Security Injections" is an ongoing collaborative project originating at Towson University [14], which integrates security topics into the undergraduate computing curriculum by developing and deploying online "security modules" for students to learn through self-study [13]. The CS courses that have been addressed with the development of security modules are primarily lower division courses with programming emphasis (CS0, CSI, CSII and CSIII). The Security Injections project emphasizes minimal intrusion and therefore promotes self-learning on the students' part without direct faculty involvement in teaching the material. Students are directed to the website by the faculty and are asked to read and learn the material themselves and to carry out assessment activities such as laboratory assignments, discussion questions, and security checklists. The only involvement mandated by the faculty is to administer security surveys before and after student exposure, direct students to the website or hand the material over to the students, and at the end take part in faculty surveys. In a workshop arranged by the Security Injections project, "getting faculty involved" was identified as one of the ways to improve the project's progress [15]. Another problem identified was that in some cases, students tended to skip materials that are online. Mostly geared at lower division programming courses, the project is being adopted by the Maryland Alliance for Information Security Assurance.

3. THE PROJECT OVERVIEW

The central themes surrounding the project objectives are student and faculty learning. The project creates and integrates relevant, adaptable, and extensible learning materials on security topics and teaching strategies for selected, required CS courses based on best practices in computer security education and pedagogy resulting in a multi-faceted instructional support system, which

- facilitates student learning in computer security by helping students understand "the big picture" and its relevancy with computer science education;
- offers ready-made instructional material to integrate security in traditional courses;
- is readily deliverable by faculty;
- is easily adaptable into CS curriculum; and
- is extendable to meet the need of the CS curriculum.

In this project, our learning model consists of assisting faculty in learning and acquiring new knowledge and skills, which will in turn, facilitate student learning (Figure 1). While the solid lines are the focus of this project, the dotted lines in the left are not.

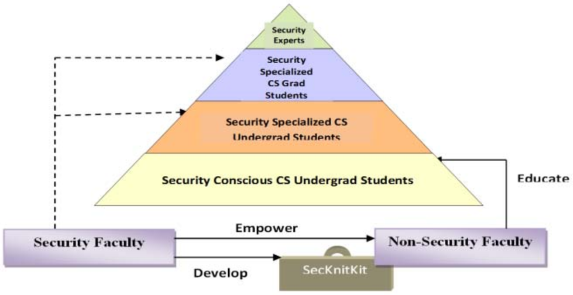
Security Experts
Security Specialized CS Grad Students
Security Specialized CS Undergrad Students
Security Conscious CS Undergrad Students
Educate
Empower
Security Faculty
Non-Security Faculty
Develop
SecKnitKit

Figure 1: SecKnitKit Model

While our primary goal is to educate CS students better in security concepts, we believe that without active involvement of the faculty learning cannot be as effective as possible. However, as noted by Bransford in [2], it is difficult for faculty to undertake the rethinking of their subject matter and to take up the challenge of learning material out of their comfort zone. It is natural for faculty to feel vulnerable by taking on such risks. While we want to involve the faculty in the learning process, we certainly do not want to create an extra burden on an already overworked faculty workforce in higher education, especially in these days of resource limitations and budget cuts. Therefore there are primarily two aspects of our approach, as shown in Figure 2: SecKnitKit instructional support system and professional development workshops.

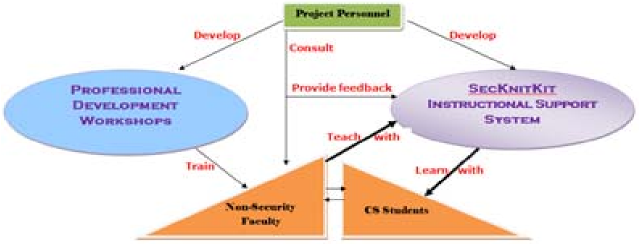

Figure 2: SecKnitKit Approach

3.1 SecKnitKit Instructional Support System

The SecKnitKit Instructional Support System is offered to any non-security faculty, allowing them to integrate relevant security topics into upper division courses as part of the course content as well as teach them about security issues with minimal effort on their part. We believe that the in-class experience is more effective than the out-of-class experience where students are left with the responsibility of sifting through materials on their own. Without over-burdening the faculty, we want to ensure that this teaching experience is as effortless as possible. The readily available instructional support system comes as a SecKnitKit toolkit (Figure 3) to be used by the faculty, which include the following:

- Instructional material in the forms of presentation slides and lecture notes where the material will include *(knowledge-centered components)*:
  - Introduction, description of the topic, and its relevancy to the main course topic;
  - Nature of the topic - whether it is a type of threat, attack,, vulnerability, or security mechanism:
    - If it is threat/attack/vulnerability, how to mitigate it;
    - If it is security mechanism, how to enforce it.
- Assessment material in the forms of (*formative and summative assessment compothounents*)**:**
  - Discussion questions in assignments to engage in problem solving and critical thinking;
  - Hands-on group active learning assignments to facilitate problem solving, active learning, critical thinking and collaborative learning (take home);
  - Quizzes and exam questions (with answers);
  - Surveys/feedback questionnaires (online).

The assessment-centered component facilitates improvement in teaching with formative assessments and assists in progress determination with summative assessments. While the knowledge centered components are delivered in class, most if not all, formative and summative assessments are designed to be online or out-of-class to save on class contact hours.

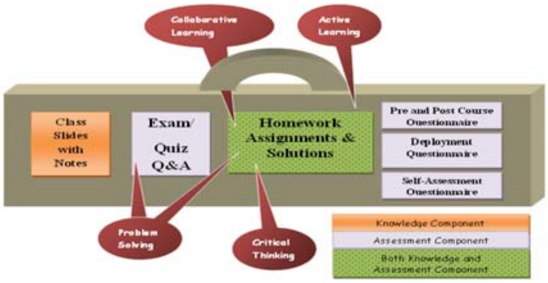

Figure 3: The SecKnitKit Toolkit

In this project, we concentrate on integrating security in four upper division courses that are common/typical in any ABET accredited CS curriculum (software engineering I, database management systems, operating systems and computer networks). With security included as integrated topics in these courses, all CS graduates will most likely experience some level of exposure to concepts of security. This also allows our modules to be adoptable to any other higher education institutions offering CS degrees. Other rationales are as follows:

- During the junior and senior years of undergraduate education, the students are more mature and have more experience, which helps them to better understand and realize the need and impact of security in all technological aspects of CS like software engineering, database management, operating systems and networks.
- Faculty who teach upper division courses are typically the ones who also teach graduate level courses and hence are more motivated and inclined to conduct research.
- For security topics related to lower division programming courses, the Security Injections project exists to offer aid to interested faculty.

For each of these courses, relevant security topics are identified using primarily Core Tier-1 IAS topics (absolutely essential topics) and some Core Tier-2 IAS topics (important foundational topics) identified in ACM/IEEE-CS CS2013 Body of Knowledge [1]; recommendations of experts in this field [4,7,10,11,16-18], computer science pedagogy, and consultation with faculty teaching these courses. The most significant challenge of content development was assessing what security material to integrate into traditional courses, considering the right balance of importance of material, relevance, and time to teach. Table 1 shows the topics in the SecKnitKit instructional modules and Table 2 shows the topics in the SecKnitKit active learning exercises.

Table 1. **SecKnitKit Instructional Module Topics**

Course	Threaded Security Concepts
ALL	Introduction to Security (basic concepts like threats, attacks, CIA model, defense-in-depth)
Software Engineering	Security risk management Security design principles Common programming errors with security implications
Operating Systems	Security design principles Common system management errors with security implications Access control Authentication Covert Channels
Database Management	Traditional security concerns in DB Security controls in DB Special security concerns in DB
Computer Networks	Security issues and controls in TCP Security issues and controls in IP

Table 2. **SecKnitKit Active Learning Exercise Topics**

Course	Security Concepts
Software Engineering	Buffer overflow attack Security problem with improper initialization Security problem with improper operand and insufficient random values
Operating Systems	Access control matrix in Windows Race condition Heap spraying Authentication in Unix
Database Management	SQL integrity control SQL access control Views SQL injection attack
Computer Networks	Man in the Middle attack with IP Spoofing Man in the Middle attack with ARP Poisoning Local cache poisoning Wireless security (secure and insecure configuration) Simple IP Spoofing

3.2 Faculty Professional Development Workshop for Non-Security Faculty

Professional development (PD) has been a well-established mechanism for enhancing educators' knowledge and skill with proven positive impacts. With the support of the SecKnitKit toolkit and effective training in adopting this approach, the learning curve for faculty was expected to be minimal and produce only a minor burden. A onetime professional development workshop for faculty was conducted midway through in the project cycle and this workshop was spilt into three parts:

- Security Awareness and Project Introduction: This was split into two components. In the first half, the faculty were introduced to the project, its goal, and the future plan. In the second half, they were introduced to current security landscape with basic concepts in IA and security, threats and protections.
- SecKnitKit Training: In the second one in the series,faculty were "walked through" the instructional and assessment materials for the courses. The instructional material was delivered in a lecture style and the faculty exercised the active learning material with guidance.
- Brain Storming: In the third, focus groups were formed for strategic discussion of issues or concerns such as:
  - General security topics of interest relevant to the courses under consideration;
  - Traditional topics in the course curriculum that can "potentially" be replaced with security topics.

It should be mentioned that since the security topics are added on to an already existing curriculum, some minor adjustments need to be made regarding the overall course structure. Careful consideration is given to the design of the integration such that total in-class delivery of security material does not exceed 1.5 to 2 lecture units of the host course. There are several alternatives to consider in the weeding out process. The traditional topics identified can be totally replaced (if appropriate), or condensed in content, and/or made available outside class through an online class portal. Throughout the workshop, local non-security faculty responsible for deploying SecKnitKit at TTU shared their experiences and insights gained by deploying SecKnitKit in TTU with the workshop participants and guided the participants in material walk throughs.

4. PROJECT IMPLEMENTATION – YEAR 1

We designed and developed the SecKnitKit toolkit during Fall of 2012. We deployed it at the TTU institutional level during Spring 2013 where we embedded security topics in four upper division courses that were taught by three faculty whose primary teaching expertise is not security. The training was conducted on a one-to-one basis.

We conducted a two day professional development workshop at the end of first academic year of the 2-year project (June 3-4, 2013) attended by 15 non-security faculty from other institutions who teach any of these courses: software engineering, operating systems, networks and database management systems. There was a call for participation and 12 out of approximately 50 applicants were carefully selected to be early adopters. They had no prior teaching experience in security, their curriculum did not offer any mandatory security courses and they had a definite interest in integrating security traditional CS courses. To defray the cost of travel, we received funding to provide each participant with a stipend of $1000.00 to attend the workshop. In addition, three faculty attended the program out of special interest without any travel reimbursement. Later two faculty from another institution volunteered to become additional early adopters.

5. PROJECT EVALUATION – YEAR 1

This year approximately 150 CS undergraduate students were exposed to SecKnitKit material through four courses at TTU in the Pilot deployment phase. These courses were Software Engineering II (CSC 4620), Operating Systems (CSC 4240), Computer Networks (CSC 4240), and Database Management Systems (CSC 4300). Pre and post surveys were administered to students and faculty associated with the courses. The purpose of these surveys was to gain preliminary data on the pilot of the SecKnitKit and related materials (including evaluation materials) before a wider dissemination to other institutions.

5.1 TTU Pilot Evaluation: Student

Overall, the results of this pilot have been very positive. However, we were able to identify some items to improve and some items that need further study. Each course was analyzed by conducting paired t-tests on the pre and post surveys of student responses. Responses were reported on a six-point scale reflecting interest and a five-point scale for awareness, knowledge and progress related to security topics. Missing data were eliminated by cases. The student survey results are reported for each course below. Due to space limitations, graphical representations of the results are not included here.

5.1.1 Results for Computer Networks

The Computer Networks course was the smallest of the four courses with only 8 students completing both the pre and post surveys. With only 8 students and only 7 degrees of freedom it was more difficult to get statistical significance between the pre and post surveys. Because of the small sample, only three variables showed statistically significant gains. The students reported significant gains on their awareness of the source of security incidents, the stand of the USA in security incidents as compared globally, and the number of computer security related jobs projected. Although generally students reported gains in knowledge in areas related to course material, (sometimes as much as a 1 point gain on a 6 point scale in the case of the abuse of ARP protocol for denial of service), due to the small sample size, these items were not statistically significant.

5.1.2 Results for Database Management Systems

The Database Management Systems course contained 43 matched pairs of student surveys. For this course, students reported significant gains in the awareness of security issues, knowledge gains in the course and reported progress in active learning and database opportunities. The students reported significant gains in awareness of the cost of security incidents, the source of security incidents, the impact of security incidents, the stand of the USA in security incidents as compared globally, and the number of computer related jobs. Students also reported significant gains in knowledge in each of the following areas: basic terms and terminology in security, threat categories, attack types, CIA model, defense in depth, security policy, traditional security concerns (CIA), security controls in DB schema, security controls with Views, and security controls in SQL. Students similarly reported significant progress on active learning and opportunities related to database security: identifying security problems in databases and how to address security in databases compared to typical previous classes. The one significant negative finding was students demonstrated less interest in learning more about security in general.

5.1.3 *Results for Operating Systems*

The Operating Systems class contained 22 matched pairs of students. Students reported a significant increase in awareness of the cost of security incidents. They reported a significant increase in knowledge of the basic terms and terminology in security, attack types, CIA model, Defense in depth, security policy, dangers of incorrect settings of process privileges, dangers of incorrect access control permissions, importance of memory protection, dangers of race conditions, importance of indivisibility, access control, authentication, use of shadow file for authentication, and covert channel detection and containment. They also reported significant progress in active learning opportunities related to network security, identifying potential security problems in networks, and how to address security in networks compared to previous classes. They demonstrated significant decrease in their agreement in security as an important issue and wanting to learn more about security in general. They were more likely to agree that they had had active learning opportunities related to network security.

5.1.4 *Results for Software Engineering II*

Software Engineering II had 16 matched pairs of students.. These students reported a significant improvement in their awareness of the cost of security incidents, the source of security incidents, the stand of the USA in security incidents as compared globally, and the number of computer security related jobs projected. They also reported the following knowledge gains: gain in knowledge in the basic terms and terminology in security, Threat categories, attack types, CIA model, defense in depth, security policy, threat modeling, risk assessment, risk mitigation, security engineering, security principles, dangers of improper configuration/initialization, dangers of buffer overflow, dangers of shared memory, dangers of poorly thought out error messages, and the importance of checking proper permissions. Students in the Software Engineering II class also reported significant progress in active learning opportunities related to computer security, identifying potential security problems, and how to think critically to improve computer security.

5.2 TTU Pilot Evaluation: Faculty

The faculty surveys for local deployment, due to the small sample size (4), were never meant to be statistically analyzed, but were meant to serve as a pilot for surveys to be administered to the larger group of faculty members in the next phase. Faculty feedback concerning SecKnitKit content in the 1st year is being taken into consideration for revising and improving for deployment in 2nd year. Also, as our motto was “minimal effort” for the non-security faculty to bring security concepts into classes, we sought feedback especially on faculty involvement and the following is the result.

- Preparation time for lectures: Approximately how much time was spent in preparing for security related lectures (not including active learning exercises), the average response was between 1 to 3 hours.
- Class time: Approximately how much time was spent in covering security topics in class the average response was between 90 to 120 minutes.
- Preparation for active learning and grading: Approximately how much time was spent in familiarizing/preparing with/for the active learning exercises?” and “Approximately how much time was spent in grading the active learning exercises, the average response was between 2 to 3 hours.

5.3 Faculty Professional Development Workshop Evaluation

The workshop was evaluated through a survey of participants. This survey included both quantitative and qualitative questions. By all indications, this workshop was successful in generating interest and awareness in non-security faculty to integrate security topics in their courses. Fifteen respondents from 13 institutions completed the workshop survey. The participants attending came from a wide range of institutions including R-1 schools such as the University of Wyoming and historically black schools such as Fisk University. Of the participants, 47% were female and 53% were male. Only 13% had a required security course for their majors. The number of years teaching computer science ranged from 1 to 30 years. Sixty percent had no computer security classes as an undergraduate while 40% had one computer security class as an undergraduate. While 73% had no computer security classes in graduate school, 7% had half a class, and 20% had one computer security class in graduate school. None of the participants had taught a computer security class previously.

All of the respondents found all of the sessions to be useful. All of the respondents also reported that the workshop provided an effective overview of the project and provided the information needed to implement the SecKnitKit All respondents similarly reported the workshop to be a good networking opportunity and indicated a willingness to share information about the workshop with other faculty members.

Similar to the students at Tennessee Tech, the faculty members across institutions participating in the workshop indicated that they had increased their awareness about computer security issues. The area with the lowest level of agreement was the area regarding awareness of funding in computer security where one faculty member slightly disagreed that the project had increased their awareness.

All aspects of the project seem to be key components in getting faculty members to incorporate computer security. The workshop seems to have accomplished its goals of providing the needed information about SecKnitKit to assist non-security faculty members in incorporating more issues related to security in their classes.

5.4 Summary of Findings and Retrospection

Overall, students reported significant gains in knowledge, awareness, and progress related to computer security in a wide variety of areas addressed by the SecKnitKit. These results are very encouraging. The primary purpose of this project is to raise awareness and importance of security in computing for CS students and faculty and the results indicate this has been successful. The one area in which the findings were not significant is the area of increasing student interest in computer security. A couple of possible explanations exist to explain why this may have occurred. First the scale for these questions was reversed from other questions on the survey. This may have confused some students. We have changed the scale of these questions for year 2 deployments so that positive and negative ratings are consistent on the survey. Another possible explanation is that most of the students in these upper division classes were seniors who likely have chosen their specific career path in CS before they were exposed to security. We have added some open-ended questions on the post survey for the current year to ask students to explain why they are (or not) interested in computer security careers and graduate studies.

Regardless, as mentioned before, indication of an increase in security awareness among CS undergraduates, and an increase in security knowledge among CS undergraduates in selected subject areas serves SecKnitKit's primary mission of raising security consciousness among CS graduates. This project has demonstrated that non-security faculty can successfully incorporate the SecKnitKit in non-security focused courses.

6. CONCLUSION

Security illiteracy is a very common problem among CS graduates entering the nation's digital workforce, which has contributed to a national cyber-infrastructure that could and should be more resilient to cyber-enemies. There is urgency in increasing the size of our security-aware workforce such that computing students enter the workforce with the knowledge needed to design and develop reliable systems - at the least –avoid poor development practices that often leave loopholes to exploit. This project focuses on integrating knowledge and understanding of security issues across multiple CS courses. Integrating the project requires active participation from the faculty; therefore faculty who do not teach security courses are provided support with adapting security topics into the existing curriculums and delivering learning material in a useful manner. It also provides an opportunity for fostering interest in security among non-security faculty, who may be motivated to become more involved in this area with teaching and/or research. The SecKnitKit toolkit is portable and accessible enough to facilitate the adoption at other higher education institutions.

We have institutionalized SecKnitKit at our university and have done it in a manner to continue even after funding ends. The successful deployment of the first phase of the project is providing innovative insights into the capabilities of threading security content into traditional CS courses. The project is in its second phase currently which involves working closely with the non TTU early adopter faculty in 13 institutions nationwide that are adopting SecKnitKit into upper division courses, using evaluation results and faculty feedback from Year 1 to improve SecKnitKit, continuing with deployment of SecKnitKit at TTU and continuing with various activities to disseminate SecKnitKit material and model findings and experiences to the CS community. All SecKnitKit material that has been developed under this project (instructional, assessment, active learning) is accessible at the project website at www.secknitkit.org as per request.

7. ACKNOWLEDGMENTS

Our sincere gratitude goes to project senior personnel William Eberle and Mike Rogers for helping with institutional deployment of this project. We are thankful for their valuable feedback, guidance, and support. Special thanks go to Suzanne Henry and Valerie Nash for ensuring compliance with TTU policies and procedures. Last but not least, we acknowledge and appreciate all the workshop attendees/early adopters, whose interest, time, and effort is instrumental in the success of this project.

8. REFERENCES

[1] ACM/IEEE-CS Joint Task Force on Computing Curricula. 2013. Computer Science Curricula 2013. ACM Press and IEEE Computer Society Press. http://ai.stanford.edu/users/sahami/CS2013/final-draft/CS2013-final-report.pdf

[2] Bransford, J. D., Brown, A. L. and Cocking, R. R. *How People Learn: Brain, Mind, Experience, and School.* National Academy Press, Washington, D.C., 1999.

[3] Computer Security Institute. The 14th Annual CSI Computer Crime and Security Survey Report. Retrieved September 3, 2013 from http://gocsi.com/survey_2009.

[4] Irvine, C., Chin, S. and Frincke, D. Integrating Security into the Curriculum. *IEEE Computer, 31*(12). pp. 25-30.

[5] Irvine, C.E., Challenges in Computer Security Education. *IEEE Software,* Sept./Oct. 1997, pp. 110-111. Retrieved September 3, 2013 from http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA484034&Location=U2&doc=GetTRDoc.pdf.

[6] Markham, S.A., Expanding Security Awareness in Introductory Computer Science Courses. in *Proceedings: Information Security Curriculum Development Conference,* (Kennesaw, GA, 2009).

[7] Mullins, P., Wynters, E., Wolfe, J., Calhoun, W., Oblitey, W., Fry, M. and Montante, R., Panel on Integrating Security Concepts into Existing Computer Courses. in *Proceedings of SIGCSE 2002*, (Covington, KN, 2002), ACM, pp. 365-366.

[8] National Security Agency and Central Security Service. 2012. National Centers of Academic Excellence in IA Education (CAE/IAE) Criteria for Measurement. Retrieved September 3, 2013 from http://www.nsa.gov/ia/academic_outreach/nat_cae/cae_iae_program_criteria.shtml

[9] National Security Agency and Central Security Service. 2013. National Centers of Academic Excellence. Retrieved September 3, 2013 from http://www.nsa.gov/ia/academic_outreach/nat_cae/index.shtml.

[10] Null, L. Integrating Security Across the Computer Science Curriculum. *Journal of Computing Sciences in Colleges,* 19(5). pp. 170-178.

[11] Perrone, L. F., Aburdene, M. and Meng, X., Approaches to Undergraduate Instruction in Computer Security. in *Proceedings of the American Society for Engineering Educational Annual Conference and Exhibition,* (2005), ASEE. Retrieved September 3, 2013 from http://www.ists.dartmouth.edu/library/116.pdf.

[12] Roberts, B., Cress, D. and Simmons, J. Towards a Security-Aware Undergraduate Computer Science Curriculum at UMBC. Retrieved September 3, 2013 from http://www.csee.umbc.edu/~cress1/ia/341-stuff/Questions-341-profs.doc

[13] Taylor, B., Kaza, S., Security Injections: Modules to Help Students Remember, Understand, and Apply Secure Coding Techniques. in *Proceedings of the 16th Annual Conference on Innovation and Technology in Computer Science,* (Darmstadt, Germany, 2011).

[14] Towson University. Security Injections. Retrieved September 3, 2013 from http://cis1.towson.edu/~cssecinj/.

[15] Towson University. Security Injections Workshop – January 2010. Retrieved September 3, 2013 from

[16] Vaughn, R., Application of Security to the Computing Science Classroom. in *Proceedings of the 31st SIGCSE Technical Symposium,* (Austin, TX, 2000), ACM, pp. 90-94.

[17] White, G. and Nordstrom, G., Security Across the Curriculum: Using Computer Security to Teach Computer Science Principles. in *Proceedings of the 19th National Information Systems Security Conference,* (Baltimore, MD, 1996), pp. 483-488.

[18] Yang, T. Computer Security and Impact on Computer Education. *Journal of Computing in Small Colleges,* 16(4). pp. 233-246.