cybersecurity education

INTEGRATING SECURITY IN THE COMPUTER SCIENCE CURRICULUM

Ambareen Siraj, Blair Taylor, Siddarth Kaza, Sheikh Ghafoor

With increased focus on the global computing infrastructure's vulnerability to cyber-attacks - the time is right for security integration across the computer science curriculum to contribute to a cyber-ready workforce. The challenges to integrating security into computer science (CS) curriculum are significant—lack of faculty to teach security, a dearth of effective teaching resources, and little room to spare in CS curriculum. This article describes an initiative that aims to develop faculty expertise in cybersecurity, provide a library of resources for security education, and build a community of CS educators to prepare computing graduates to meet current and future cybersecurity challenges.

Securing the computing infrastructure (commonly referred to as 'cybersecurity') has critical political, military, economic and social implications for all nations. Lack of information assurance and security in deployed computer software costs businesses and taxpayers severely every year. Producing a cyber-prepared workforce requires pervasive action—security across the curriculum—to prepare all computing graduates to meet the current and future cybersecurity challenges. To address this, the recent ACM/IEEE-CS Curricula 2013 [1] has designated Information Assurance and Security (IAS) as a new cross-cutting knowledge area and the NSA/DHS Center of Academic Excellence (CAE) certification criteria require the integration of security concepts in computing curriculum. However, the challenges to integrating security into computer science (CS) curriculum are significant for most institutions—lack of faculty to teach security, lack of teaching resources, and little room to spare in CS curriculum.

Funded by the National Science Foundation (NSF), the Cyber Workshops: Resources and Strategies for Teaching Cybersecurity in Computer Science (CReST) project aims to empower CS faculty to integrate security concepts in their curriculum. This project builds upon the success of previously funded NSF cybersecurity education projects by using experienced leaders from the cybersecurity education community to launch a series of regional cyber workshops for developing faculty expertise in cybersecurity, arm faculty with cybersecurity teaching resources, and build a community of cybersecurity educators. Specifically, this project has two thrusts:

1. *Engage the existing cybersecurity faculty community* in conducting workshops and use their regional collaborations to motivate computer science faculty to attend workshops and include security in their classes. This is accomplished by appointing faculty advocates nationwide and providing support to organize regional cybersecurity workshops. Cybersecurity faculty are invited and supported to present well-tested cybersecurity teaching resources in these workshops to provide a diversity of material, experience, and course exposure.
2. *Support computer science faculty to include appropriate security concepts* in their courses. Workshop participants include faculty from two-year and four-year institutions teaching both security and non-security courses in computer science. Using a well-honed workshop model—which includes security integration strategies, incentives to initiate and sustain adoption, research-based practices, and which provides a wealth of field-tested teaching resources from effective projects—this support is designed to empower all CS faculty to teach cybersecurity and have a lasting impact on cybersecurity education.

The CReST project includes resources from Security Knitting Kit [11], Security Injections@Towson [13], and several other well-established projects, including EDURange [2] and SEED [3]. The Security Knitting Kit project includes security educational resources for integrating security into upper-level CS courses (Database, Operating Systems, Networks, Software Engineering). The Security Injections@Towson project includes modules for lower-level CS courses (Computer Literacy, CS0/CS1/CS2 programming with C++/Java/Python/Pseudo-Code) and web development. Together, the two projects provide over fifty teaching modules that span ten courses in the CS curriculum.

Security Knitting Kit resources can be found at www.secknitkit.org

Security Injections@Towson modules can be found at www.towson.edu/securityinjections

cybersecurity education

Integrating Security in the Computer Science Curriculum

SECURITY KNITTING KIT (SECKNITKIT) PROJECT

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 has developed, deployed, and disseminated a multi-faceted out-of-the-box instructional support system to empower non-security faculty. (See SecknitKit Toolbox in Figure 1).

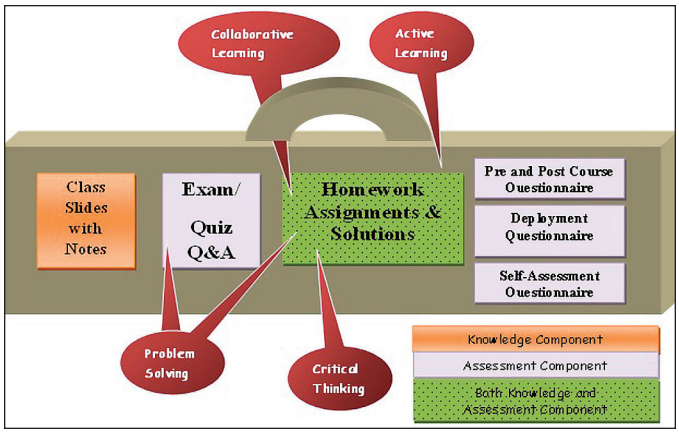

Figure 1: SecKnitKit Toolbox here

This project enables faculty to weave effectively relevant security topics into four upper division courses that are common/typical in any ABET accredited CS curriculum (software engineering, database management systems, operating systems and computer networks). SecKnitKit modules can also be adapted to other higher education institutions offering CS degrees. Table 1 shows the topics in SecKnitKit instructional modules and active learning exercises.

In year 1, CS undergraduate students received SecKnitKit material through four courses at Tennessee Tech University (TTU) in the pilot deployment phase [7]. Three non-security TTU faculty members taught these courses and provided feedback that showed that they were able to deploy SecKnitKit in their courses with minimal effort. Overall, TTU students reported significant gains in knowledge, awareness, and progress related to computer security in the wide variety of areas addressed by the SecKnitKit.

During the summer of 2013, fall 2013 and spring 2014, the SecKnitKit materials were disseminated among U.S. institutions. Ten institutions[1] completed responses on both pre- and post-surveys. Survey responses from faculty indicate that minimal effort was required in integrating resources in their courses (summary responses here):

- *Approximately how much time have you spent preparing for security related lectures (not including active learning exercises)?* Majority (70%+) reported between 2 to 5 hours.
- *Approximately how much time have you spent covering security topics in class?* Majority (70%+) reported between 1 to 3 hours of instructional time.

[1] University of Wyoming, James Madison University, Murray State University, College of St. Scholastica, Fairmont State University, Middle Tennessee State University, University of Central Arkansas, University of North Carolina at Wilmington, University of North Texas and Tennessee Tech University.

TABLE 1. SecKnitKit INSTRUCTIONAL MODULES WITH ACTIVE LEARNING EXERCISES

Course	Threaded Security Concepts	Active Learning Exercises
ALL	Introduction to Security (basic concepts likes threats, attacks, CIA model, defense-in-depth)	
Software Engineering	Security risk management Security design principles Common programming errors with security implications	Buffer overflow attack Security problem with improper initialization Security problem with improper operand and insufficient random values
Operating Systems	Security design principles Common system management errors with security implications Access control Authentication Covert Channels	Access control matrix in Windows Race condition Heap spraying Authentication in Unix
Database Management	Traditional security concerns Special security concerns in Database Security controls in Database Management Systems	SQL integrity control SQL access control Views SQL injection attack
Computer Networks	Security issues, controls in TCP Security issues, controls in IP	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

- *Approximately how much time have you spent familiarizing with/ preparing for the active learning exercises?* Majority (70%+) reported between 2 to 4 hours.
- *Approximately how much time have you spent grading the active learning exercises?* Majority (85%+) reported between 1 to 3 hours.

After participating in this project, all faculty members reported an awareness of the issues that relate to the project varying from somewhat aware to very much aware. All but one faculty member agreed that after participating in this project, they felt more comfortable teaching the CIA model and defense in depth. All faculty members agreed that they would like to teach more about computer security.

Overall, the results indicate that this project has been a success. Although funded for only local institutionalization, there has been successful implementation across U.S. Students reported significant gains in knowledge, awareness, and progress related to computer security in a wide variety of areas addressed by the SecKnitKit. They also reported an exceptional level of interest in both learning more about computer security and careers in security fields. Similarly, faculty reported both an awareness and comfort with computer security topics from participating in this project. All components of the project were rated important to faculty in order to facilitate incorporation of security topics. All faculty were interested in teaching more about security and most faculty were interested in security related research and networking with others on security related topics. This project has demonstrated that non-security faculty can successfully incorporate the SecKnitKit in non-security focused courses.

SECURITY INJECTIONS@TOWSON PROJECT

The 'Security Injections@Towson' project includes over 40 security injection modules that target key security concepts including integer overflow, buffer overflow, and input validation for CS0, CS1, and CS2 (an example Security Injections module is shown in Figure 2) . By targeting courses that are often neglected in security education, the modules emphasize learning robust programming practices early. A representative list of the current security injections library appears in Table 2. An example module is shown in Figure 2 along with a security checklist (Figure 3) that students complete to locate vulnerabilities in code.

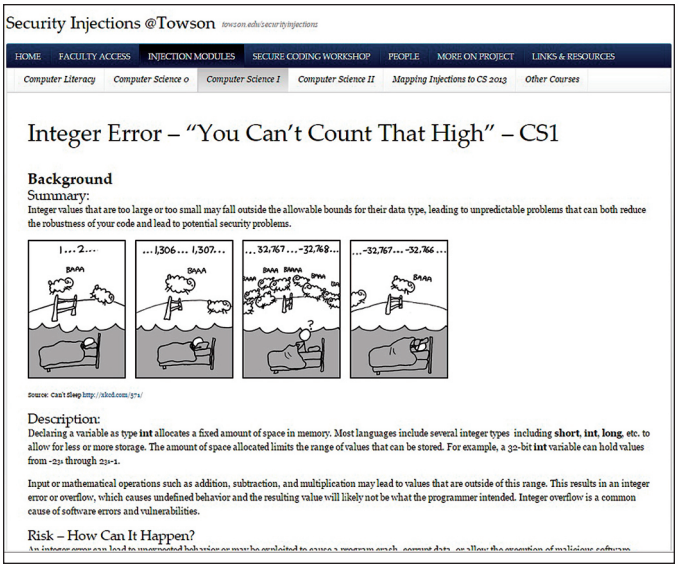

Figure 2: CS0 Integer Error Security Injection module

Since 2008, thousands of students and over three hundred faculty have used the modules across 151 diverse institutions, including two-year and four-year colleges and several high schools. The Security Injection modules have found to be effective at improving security awareness and the results have been consistent across ethnicity, gender, and student standing [4,6,7,8].

The formal project evaluation used a variety of instruments, including surveys, random sampling of assignments, qualitative inputs from faculty, controlled experiments in classrooms, and institutional quantitative data. Independent consultants in both technical content and pedagogical assessment provided additional feedback. Results show that security injections help students retain, comprehend, and apply secure coding concepts in the CS0 and CS1 introductory courses [4,8] and increase students' security awareness in all computing courses [4,8,9,10,14]. Additionally, the enhanced version of the modules—that include auto-grading, checkpoint questions and instant-feedback—led to an increase in student engagement (unpublished results). Feedback from instructors indicates higher student and instructor interest.

TABLE 2. THE SECURITY INJECTIONS LIBRARY

Injection module	Course	Language			
		C++	Java	Python	Pseudo code
Integer Overflow	CS0, CS1, CS2	X	X		X
Input Validation	CS0, CS1, CS2	X	X	X	X
Buffer Overflow	CS0, CS1, CS2	X	X	X	X
Cross-site scripting	Web Development	PHP, Ruby on Rails			
SQL Injections Intro	Database (CIS & CS)	SQL			
SQL Injections Advanced	Database (CIS & CS)	SQL			
Wi-Fi Security	Networking	NA			
Physical Security	Networking	NA			
Phishing	Computer Literacy	NA			
Passwords	Computer Literacy	NA			
Cryptography	Computer Literacy	NA			

CREST PROJECT APPROACH

The primary target group for professional development in the CReST project is CS faculty whose main teaching/research expertise is not security. The goal is to empower

cybersecurity education

Integrating Security in the Computer Science Curriculum

```
#include <iostream>
#include <limits>
using namespace std;
int main(void)
{
    int i;
    int j;

    cout << "For this compiler: " << endl;
    cout << "integers are: " << sizeof (int) << " bytes " << endl;
    cout << "largest integer is " << INT_MAX << endl;
    cout << "smallest integer is " << INT_MIN << endl;

    cout << "Input two integer values " << endl;
    cin >> i >> j;

    cout << endl << "You entered the following values: " << endl;
    cout << "integer " << i << " " << j << endl;

    int result = i * 10;
    cout << "Your number times ten is " << result << endl;
    result = i + j;
    cout << "The sum of your numbers is " << result << endl;
    result = i * j;
    cout << "The product of your number is " << result << endl;

    return 0;
}
```

Vulnerability: *Integer Errors* **Course:** *CSO*	
Check each line of code	**Completed**
1. Click each declaration of an integer variable.	✓
For each variable from 1:	
2. Click all input operations that assign values to the variable.	✓
3. Click all mathematical operations involving the variable.	✓
4. Click all assignments made to the variable.	✓
Highlighted areas indicate vulnerabilities!	

Figure 3: Security Checklist in the CS0 Integer Error Security Injection Module

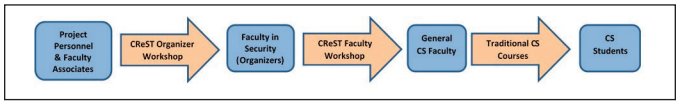

Figure 4: CReST Model here

them to integrate relevant security topics into their traditional CS courses seamlessly and effectively. To ensure widespread dissemination, four CReST faculty workshops will be offered across all regions in the United States over the duration of the project.

In the CReST model (Figure 4), security faculty members serve as either Organizers or Associates.

1. *Faculty Organizers* are comprised of faculty in security who recognize the need for security to be integrated in CS curriculum and are willing to serve as organizers for the CReST faculty workshop at their institution.
2. *Faculty Associates* consist of principal investigators of successful security education projects willing to share their materials with other CS faculty. For the associates, the benefits of sharing are:

- wider dissemination of their work in the CS community outside of their local institution;
- networking opportunity with peers with similar interest; and
- sustainability of their work beyond the funded project cycle.

Assistance and resources offered by associates will enrich CReST faculty workshop content and online shared repository of security education materials. The CReST project seeks:

- CS educators interested in including cybersecurity in their programs to provide training, resources, and support;
- security faculty who have developed and implemented security educational modules in the classroom to serve as faculty associates in an upcoming CReST workshop; and
- faculty from institutions with a proven track record in cybersecurity education/research to host a regional CReST workshop at the institution.

> *Together, we will increase the number of security-aware undergraduate CS students to address global cybersecurity challenges.*

CONCLUSION

With increased focus on the global computing infrastructure's vulnerability to cyber-attacks and recent educational initiatives designed to produce a cyber-ready workforce the time is right for security integration across the computer science curriculum. The CReST project provides an essential step towards transforming security education by launching a series of cyber-workshops across U.S. to develop faculty expertise in cybersecurity, arm them with cybersecurity teaching resources, and build a community of cybersecurity educators. By incorporating several effective cybersecurity education materials that target the entire undergraduate program this project will ensure that students are taught security principles from the first CS course. By using a workshop model that involves faculty as the source of change, CReST will exponentially increase the number of faculty who require their students to practice security.

The first CReST workshop will be held at George Washington University on July 13-14th, 2015. Details can be found at www.crest4cs.org. **Ir**

References

[1] ACM and IEEE-CS. Computer Science Curricula 2013; www.cs2013.org. Accesses 2015 April 23.
[2] Boesen, S., et al. "EDURange: Meeting the Pedagogical Challenges of Student Participation in Cybertraining Environments", In *Proceedings of the 7th Workshop on Cybersecurity Experimentation and Test*. 2014.
[3] Du, W. "The SEED Project: Providing Hands-on Lab Exercises for Computer Security Education." *IEEE Security and Privacy Magazine*, September/October, 2011.
[4] Kaza, S., et al. "Injecting Security in the Curriculum – Experiences in Effective Dissemination and Assessment Design." In *Proceedings of the 14th Colloquium for Information Systems Security Education*, Baltimore, MD 2010.

[5] Pothamsetty, V. "Where Security Education is Lacking." In *Proceedings of the 2nd Annual Conference on Information Security Curriculum Development* - InfoSecCD'05, ACM Press (2005).
[6] Raina, S., Kaza, S. and Taylor, B. "Segmented and Interactive Modules for Teaching Secure Coding: A Pilot Study." In *Proceedings of the 1st International Conference on e-Learning e-Education and Online Training (eLEOT)*, Bethesda, MD, USA; September 18-20, 2014.
[7] Siraj, A. et al. "Empowering Faculty to Embed Security Topics into Computer Science Courses." In *Proceedings of the 19th Annual Conference on Innovation and Technology in Computer Science Education (ITICSE)*, Uppsala, Sweden, 2014.
[8] Taylor, B. and 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 Education* (ITICSE), Darmstadt, Germany, 2011.
[9] Taylor, B. and Azadegan, S. "Using Security Checklists and Scorecards in CS Curriculum." *National Colloquium for Information Systems Security Education*, (2007): 4-9.
[10] Taylor, B. and Azadegan, S. "Moving beyond security tracks: integrating security in cs0 and cs1." *ACM SIGCSE Bulletin*, ACM (2008): 320–324.
[11] Tennessee Tech University. Security Knitting Kit; www.secknitkit.org Accessed 2015 April 20.
[12] Tennessee Tech and Towson University; CReST. www.crest4cs.org Accessed 2015 April 20.
[13] Towson University. Security Injections; www.towson.edu/securityinjections Accessed 2015 April 20.
[14] Turner, C., Taylor, B., and Kaza, S. "Security in Computer Literacy- A Model for Design, Dissemination, and Assessment." In *Proceedings of the 41st SGCSE Technical Symposium on Computer Science Education*, Dallas, TX 2011.

AMBAREEN SIRAJ
Computer Science, Tennessee Tech University
P.O. Box 5101 Cookeville, Tennessee 38505 USA
asiraj@tntech.edu

BLAIR TAYLOR
Computer and Information Sciences, Towson University
8000 York Road, Towson, Maryland 21252 USA
btaylor@towson.edu

SIDDHARTH KAZA
Computer and Information Sciences, Towson University
8000 York Road, Towson, Maryland 21252 USA
skaza@towson.edu

SHEIKH GHAFOOR
Computer Science, Tennessee Tech University
P.O. Box 5101 Cookeville, Tennessee 38505 USA
sghafoor@tntech.edu

DOI: 10.1145/2766457 © 2015 ACM 2153-2184/15/06 $15.00