

TOWSON UNIVERSITY
COLLEGE OF GRADUATE STUDIES AND RESEARCH

INCREASING LEARNING AND ENGAGEMENT IN CYBERSECURITY THROUGH
SEGMENTED AND INTERACTIVE MODULES

by
Sagar Raina

A Dissertation
Presented to the faculty of
Towson University
in partial fulfillment
of the requirements for the degree
Doctor of Science

August 2016
Towson University
Towson, Maryland 21252


© 2016 by Sagar Raina

All Rights Reserved

TOWSON UNIVERSITY
COLLEGE OF GRADUATE STUDIES AND RESEARCH

DISSERTATION APPROVAL PAGE

This is to certify that the dissertation prepared by Sagar Raina, entitle "INCREASING LEARNING AND ENGAGEMENT IN CYBERSECURITY THROUGH SEGMENTED AND INTERACTIVE MODULES", has been approved by this committee as satisfactory completion of the requirement for the degree of Doctor of Science in Information Technology in the department of Computer and Information Sciences.



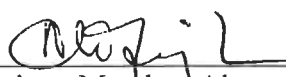
Co-Chairperson, Dissertation Committee, Siddharth Kaza
Date 8/1/16



Co-Chairperson, Dissertation Committee, Blair Taylor
Date 8/1/16



Committee Member, Gabriele Meiselwitz
Date 8/1/16



Committee Member, Alexander Wijesinha
Date 8/1/16



Dean, Graduate Studies
Date 8-5-16

ABSTRACT

INCREASING LEARNING AND ENGAGEMENT IN CYBERSECURITY THROUGH SEGMENTED AND INTERACTIVE MODULES

Sagar Raina

Cybersecurity is a global crisis. Continuously increasing cyber threats and attacks have lead the United States to take several initiatives to produce skilled cybersecurity workforce professionals. One such initiative is the introduction of cybersecurity education in schools. Since a majority of cybersecurity problems are attributed to software vulnerabilities, there is a need for teaching secure coding and computer security concepts to students using effective cybersecurity learning modules. Learning intervention based on modules are common in computer science education. Some cybersecurity learning modules have been developed, including the Security Injections @Towson cybersecurity modules. Learning modules that present a large amount of content on a single web page in a linear format may lead to pedagogical issues including - 1) content skipping, and 2) lower student engagement and learning. Addressing these issues in web-based learning modules is critical at a time when module-based pedagogical approach is widely adopted by instructors in academia and industry. This research presents a theoretical framework that uses the e-learning design principles of segmentation and interactivity to address these issues; describes a system built on this framework; and tests its effectiveness through quasi-experimental studies using the

Security Injections @Towson cybersecurity modules in computer literacy, Computer Science 0 (CS0) and Computer Science I (CS1) courses.

A total of four studies compare linear modules and segmented-interactive modules using the two group control group experimental design in the following order - 1) student engagement evaluations using post surveys in spring 2014; 2) student learning (retention of knowledge and ability to apply knowledge) evaluations using pre-survey, post-survey in fall 2014 and spring 2015, 3) students' content skipping evaluations using an eye-tracking in fall 2015 and spring 2016; and 4) usability evaluations using surveys in fall 2015. A significant increase in student engagement ($p < 0.05$), ability to apply knowledge (0.05) and students' content reading ($p < 0.05$) was demonstrated by students using segmented-interactive modules compared to linear modules. The segmented-interactive modules were found to be significantly ($p < 0.05$) more usable than the linear modules. In addition, students indicated higher interest towards segmented-interactive modules.

Table of Contents

List of Tables.....	viii
List of Figures	ix
1. Introduction.....	1
2. Literature Review	8
2.1 Content skipping in hypertext documents.....	8
2.1.1 <i>Prior knowledge & Hypertext structure</i>	10
2.1.2 <i>Learners' Interest</i>	11
2.1.3 <i>Complexity of Concept</i>	11
2.2 Interactivity and Engagement.....	12
2.2.1 Interactivity	14
2.2.1.1 Dialoguing - Assessment	14
2.2.1.2 Dialoguing - Feedback.....	15
2.2.1.3 Controlling the Presentation	17
2.2.2 <i>Approaches to Learning – Surface & Deep</i>	17
2.3 Learning in Interactive versus non-interactive systems	18
2.4 Eye tracking and Reading Research.....	19
2.5 Usability and Learning	22
2.6 Theoretical Framework	24
3. Method.....	27
3.1 Research Questions	27
3.2 Research Design.....	28
3.3 Sample.....	29
3.4 Instruments	31
3.4.1 <i>Pre and Post Software Security Survey</i>	31
3.4.2 <i>Pre and Post Computer Literacy Survey</i>	33
3.4.3 <i>Student Engagement Survey</i>	35
3.4.4 <i>Module Usability Survey</i>	36
3.4.5 <i>Eye-tracking Apparatus</i>	36
3.5 Learning Modules	39

3.6	Procedure.....	40
3.7	Hypotheses	43
3.8	Data Analysis.....	46
3.9	Limitations and Assumptions	51
3.10	Institutional Review Board	52
3.11	Summary	52
4.	System Implementation	53
4.1	Module Design	53
4.1.1	<i>Security Injections@Towson Modules</i>	54
4.2	System Development.....	58
4.3	Summary	68
5.	Results.....	70
5.1	Can the use of learning modules with segmentation reduce content skipping as compared to linear modules?	70
5.2	Can the use of learning modules with segmentation and interactivity increase student engagement as compared to linear modules?	72
5.3	Can the use of modules with segmentation and interactivity increase student learning as compared to linear modules?.....	74
5.4	Are learning modules with segmentation and interactivity significantly more usable than linear modules?	80
5.4.1	Students' Comments	86
6.	Conclusions and Discussion	88
	Appendices.....	92
	Appendix A – Institutional Review Board Documents.....	93
	Appendix B – Assessment for Student Learning	94
	APPENDIX 1: Pre-Survey CS0, CS1	94
	APPENDIX 2: Post survey CS0, CS1, CS2	99
	APPENDIX 3: Pre Survey Computer Literacy.....	107
	APPENDIX 4: Post Survey Computer Literacy	113
	APPENDIX 5: Module Usability Survey	122
	List of References	126
	Curriculum Vitae.....	135

List of Tables

Table 1: Summary of Research Methods	6
Table 2: Code Segments (Ability to apply).....	32
Table 3: Survey questions (Software Security Awareness).....	33
Table 4: Survey Questions (Phishing Awareness).....	34
Table 5: Sample e-mail to identify Phishing (Ability to apply).....	34
Table 6: Student engagement survey questions	35
Table 7: Usability Questionnaire	38
Table 8: Score evaluation matrix for ability to phishing knowledge	49
Table 9: Results of individual engagement survey questions	73
Table 10: Mean usability scores for individual questions in a survey	82
Table 11: Mean scores for individual questions in a usability survey for integer overflow, input validation, buffer overflow in CS0 and CS1 (control and treatment).....	86

List of Figures

Figure 1: Theoretical framework of segmentation and interactivity.....	4
Figure 2: Reading-skimming detection algorithm	22
Figure 3: Eye-tracker mounted on a 17" inch monitor	37
Figure 4: Calibration process steps.....	41
Figure 5: Data processing in Eye-tracking.....	47
Figure 6: Fixations detected above the text line from eye-tracker software.....	48
Figure 7: Fixations aligned on the text line using EyeMap	48
Figure 8: Segmentation & Interactivity Module Design.....	54
Figure 9: Original Security Injections Linear Module Format	56
Figure 10: Mapping cognitive levels in Blooms Taxonomy to enhanced learning modules	57
Figure 11: Django MVC architecture	58
Figure 12: MCQ Html code snippet.....	60
Figure 13: MCQ xml snippet	60
Figure 14: Constructed Response HTML Code snippet	61
Figure 15: Constructed Response XML code snippet a) Bold: shows elaborated feedback b) Gray: shows regular expression.....	61
Figure 16: Systems development process	62
Figure 17: List of security injection modules	63
Figure 18: Enhanced security injection module (Background section)	64
Figure 19: Security injection module (code responsibly section).....	65
Figure 20 : Security injection module (Laboratory assignment section).....	66
Figure 21: Enhanced Security injection modules(Discussion question section)	67
Figure 22: Enhanced security injection module (auto-graded security checklist).....	68
Figure 23: Enhanced security injections module (Elaborative explanation)	68
Figure 24: Average reading scores in control and treatment groups.....	71
Figure 25 : Average reading depth in control and treatment groups.....	72

Figure 26: (a) Average student engagement score in treatment and control group (b) Average student engagement score between males and females in treatment and control group (c) Average student engagement score between ethnic groups in treatment and control group	73
Figure 27: Pre-survey and post-survey scores in the control and treatment groups for security-coding awareness	75
Figure 28: Pre-survey and post-survey scores in the control and treatment groups for general software security awareness	76
Figure 29: Pre-survey and post-survey scores in the control and treatment groups for general phishing awareness.....	78
Figure 30: Post-survey scores in the control and treatment groups for ability to identify security vulnerability in three code segments	79
Figure 31: Post-survey scores in the control and treatment groups for ability to identify phishing in an email	80
Figure 32: Mean scores for overall usability in control and treatment group.....	81
Figure 33 : Mean usability scores in treatment and control group in CS0 and CS1 (integer Overflow).....	83
Figure 34:Mean usability scores in treatment and control group in CS0 and CS1 (input validation)	84
Figure 35: Mean usability scores in treatment and control group in CS0 and CS1 (Buffer Overflow).....	85

1. Introduction

Cybersecurity is a crisis in the United States as cyber-threats are continuously evolving and there is a need of people with the requisite technical skills to deal with these threats (CSIS, 2010; Rowe, Lunt, & Ekstrom, 2011). The majority of the computer security problems are because of software vulnerabilities and the number of software vulnerabilities that have been growing both in numbers and types in recent years.

In response to this cybersecurity crisis, there is a need of teaching secure coding principles and important information security related topics to computing and non-computing undergraduate majors respectively, through effective cybersecurity learning materials. Several cybersecurity learning materials have been developed across the United States (NICCS, 2016). The use of web-based learning modules is common among educators.

A learning module is a self-contained teaching material with a well-defined structure including information about the topic to be taught, a sequence of learning activities and evaluations to assess student learning (Robinson & Crittenden, 1971). Despite of several potential benefits including ease of access and use; learning modules with large content presented on a single webpage in a linear format involve several learning issues including content skipping and lower engagement leading to lower student learning (J. S. Taylor, 2000).

One such example is web-based security injection learning modules, developed

by Towson University. These learning modules target key secure coding concepts including integer error, buffer overflow, input validation in various programming languages, for Computer Science 0 (CS0), Computer Science 1 (CS1), and Computer Science 2 (CS2); and general security concepts, such as phishing, passwords, and cryptography for use in Computer Literacy. The modules follow a linear format on a single webpage and are developed on the cognitive learning principles of Bloom's taxonomy which adopts a uniform structure. Each module begins with a background section to describe the problem with examples, followed by Code Responsibly (includes methods to avoid security issues), a laboratory assignment with a security checklist, and discussion questions sections (B. Taylor & Kaza, 2011a). In order to complete the module, students have to read the background section followed by code responsibly section and then complete the laboratory assignment with security checklist and discussion questions.

In over six years of dissemination to over 150 institutions, one of the issues that were observed by instructors was that students tended to skip content and proceed directly to lab exercises. Skipping parts of text may lead to lose important information, and result in shallow reading, less concentration and attention towards content , and poor learning (Duggan & Payne, 2011; Liu, 2005; Rudestam & Schoenholtz-Read, 2010).

Previous research attributes content skipping in a large web-based hypertext pages to – 1) scrolling up and below the document, 2) flexibility to click any of the hyperlinks that links within or outside the page to gain knowledge and thus losing context of the original text, 3) reading strategy adopted by readers which determines what to read and what to skim, 4) perception of the large length of document among readers (DeStefano &

LeFevre, 2007; Hornbæk & Frokjaer, 2003; Lawless, Brown, & Mills, 2003) . In addition, another factor contributing towards content skipping is lack of engagement with the content. In order to maximize the effectiveness, learning modules should be designed to ensure – 1) students read the content without much skipping and 2) increase student engagement.

In this research, first, a theoretical framework grounded in e-learning design theories, aiming to reduce content skipping, increase engagement and learning, is set; second, using security injections @Towson cybersecurity modules, that follow a linear format and involve learning issues, is enhanced to incorporate the proposed e-learning design principles; third, empirical studies are conducted to test the effectiveness of enhanced learning modules as compared to linear modules on students’ content skipping, student engagement, student learning and module usability.

This research proposes to incorporate e-learning design principles of segmentation and interactivity to reduce content skipping, increase student engagement and learning in linear modules. Segmentation means, breaking large module content into smaller sections and presenting each section one at a time (Al-Samarraie, Teo, & Abbas, 2013; Clark & Mayer, 2011; Hessler & Henderson, 2013; Moreno & Mayer, 2007; Wu, Tennyson, & Hsia, 2010). Interactivity is, “responsiveness to the learner’s actions during learning” (Moreno & Mayer, 2007). The theoretical framework (see **Figure 1**) proposed in this research hypothesizes – 1) segmentation of content will reduce content skipping (Protopsaltis & Bouk, 2005; Tseng, 2008), 2) reduced content skipping will increase learning (Rudestam & Schoenholtz-Read, 2010), 3) interactivity will increase engagement (Zhang, 2010) and 4) an increase in engagement will increase

learning(Zhang, 2010) .

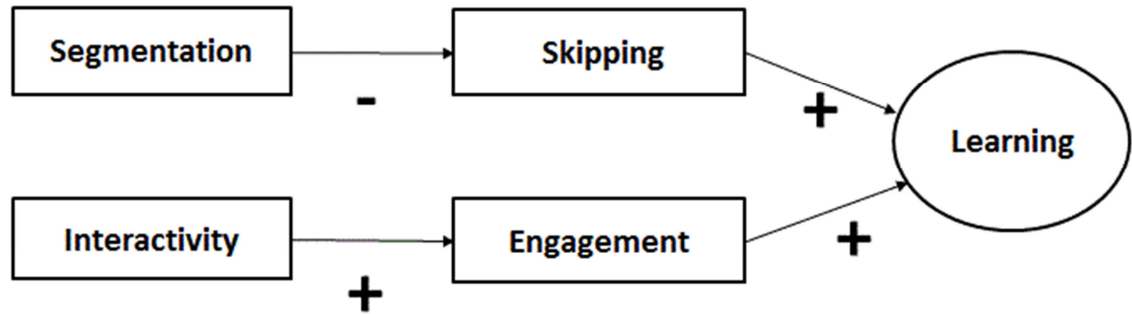


Figure 1: Theoretical framework of segmentation and interactivity

In order to test the effectiveness of the proposed solution, this research examines the following research questions:

RQ1: Can the use of learning modules with segmentation reduce content skipping as compared to linear modules?

RQ2: Can the use of learning modules with segmentation and interactivity increase student engagement as compared to linear modules?

RQ3: Can the use of modules with segmentation and interactivity increase student learning as compared to linear modules?

RQ4: Are learning modules with segmentation and interactivity significantly more usable than linear modules?

This research study began by first enhancing original (linear) security injection modules to incorporate e-learning design principles of segmentation and interactivity. A total of four studies, aimed at addressing each research question, were conducted over a

period of four semesters. In each study, a two group control group experimental design approach was adopted. The participants in these studies were undergraduate students in CS0, CS1 and Computer Literacy courses at Towson University. Each study was conducted during the laboratory sessions. The control group administered linear modules and experimental group enhanced (segmented-interactive) modules. Instruments in each study varied – a pre-test post-test approach was used to examine student learning to answer RQ3; a post-test only approach was used to examine student engagement and module usability to answer RQ2 and RQ4; and an eye-tracking approach was used to examine students' content skipping to answer RQ1 (See **Table 1**).

The research was conducted with acknowledgment of the following limitations:

1. The selection of students was limited to students enrolled in computer science core (CS0) and computer literacy courses at Towson University during the fall 2014, spring and fall 2015, and spring 2016.
2. The sample selection was not randomized, due to inherent limitation of education research set up.
3. The courses identified for the study were taught by several instructors; the study was limited due to possible variations in teaching style and syllabi of the instructors.
4. This research used student survey instruments. Although it is assumed that students answered questions to the best of their ability, this study was limited due to the accuracy of this assessment.

Research Question	Research Design	Instruments	Dependent variables	Semester
RQ1	Two group control-experimental	Eye-tracking	reading scores, reading depth	Fall 2015, Spring 2016
RQ2	Post only two control group-experimental group	User Engagement Scale (Survey)	Student engagement scores	Fall 2014
RQ3	Pre-test Post-test control group – experimental group	Security awareness survey, Ability to apply secure coding survey	General security awareness scores, secure coding awareness scores, ability to apply secure coding knowledge scores	Spring 2015
RQ4	Post only two control group-experimental group	Usability survey	Overall usability scores	Fall 2015

Table 1: Summary of Research Methods

5. The research used eye-tracking device to examine students' content skipping. The device being economically expensive, only one device could be used, resulting in less students participating in the select study due to time constraints.

The following chapters of this thesis describe literature review, research methods, results, conclusion and the future work.

2. Literature Review

Web-based learning modules present content using hypertext. Individuals reading hypertext have a tendency to skip the content (J. S. Taylor, 2000). This chapter begins with the review- why hypertext readers skip the content, followed by, the ways to reduce content skipping including segmentation and interactivity. Thereafter, a review on learning outcomes in interactive versus non-interactive systems with similar content is discussed, followed by, reading research in eye-tracking. This chapter ends with a review of literature on usability of e-learning systems.

2.1 Content skipping in hypertext documents

Hypertext is defined as a document that contains variety of media resources such as text, audio, video, graphics, presented in a non-linear fashion unlike printed textbooks. The media resources, in a hypertext, may link to other documents or within the same document. Thus readers are flexible to click any of the links and acquire knowledge (Altun, 2000; DeStefano & LeFevre, 2007; Liu, 2005). This flexibility to click any of the hyperlinks that links within or outside the page to gain knowledge lose context of the original text, thus leading to content skipping.

Acquiring knowledge from reading hypertext need not to be in a specific order. The readers, while reading hypertext, may adopt reading strategies depending upon their set goals or interest (DeStefano & LeFevre, 2007; Lawless et al., 2003; Protopsaltis & Bouk, 2005). These strategies allow reader to decide what to read and what to skim (Protopsaltis & Bouk, 2005).

The decision to skip text also depends on the amount of content presented. Huge

volume of hypertext presented on a screen tend readers to skip the text in comparison to less amount of hypertext (Duggan & Payne, 2011). (Hornbæk & Frokjaer, 2003) analyzed the reading behavior in long documents and observed that readers skip some part of the document as they scroll up and down. (Beymer, Russell, & Orton, 2005) studied the reading behavior in wide versus narrow paragraphs and found more skipping in narrow paragraphs due to increased height, and perception of lengthy material encourages skipping ahead.

During text skimming, readers read the first half of the paragraphs and if they think the information gain is low, they skip the rest of the paragraph and start reading the next (Duggan & Payne, 2011). During this process, readers might skip important content (Duggan & Payne, 2011; Protopsaltis & Bouk, 2005). In addition, reading selectively results in less in-depth reading, less concentration and less attention towards the content (Liu, 2005). Thus, skipping content in-turn results negatively towards knowledge consumption (Dyson & Haselgrove, 2000).

In a study by (Liu, 2005), 78 percent of the participants reported they read more selectively because of huge amount of information available on web. The process of selecting a content to read, in a hypertext, requires readers to make decision. Making this decision induces cognitive overload in a reader leading to content skipping (DeStefano & LeFevre, 2007; Protopsaltis & Bouk, 2005). There are several other factors that might lead to content skipping in a reader which includes - lack of prior or domain knowledge about the content, complexity of concept, structure of the content, lack of interest or motivation to read content (Al-Samarraie et al., 2013).

2.1.1 Prior knowledge & Hypertext structure

Prior knowledge is knowledge about the subject a learner has already possessed, whereas domain knowledge is the knowledge about the specific field of study a learner has acquired (Jetton & Alexander, 2001). The readers have prior or domain knowledge connect with easily with the content and do not skip the text, whereas, the lack of prior knowledge about the subject might lead readers to skip the content.

Hypertext structure drives from how the content is structured and presented to the user (Cangoz & Altun, 2012). Well-structured hypertext is the presentation of the content in a specific format on a computer screen. Well-structured content has high coherences between successive sentences or paragraphs enabling learners to read, understand and synthesize the concept easily and allow them to stay oriented with the text, therefore less skipping (Al-Samarraie et al., 2013; Niederhauser, 2008; Protopsaltis & Bouk, 2005; Shapiro, 1998).

A study conducted by (Shapiro, 1998) compared expert learners, who have high prior domain knowledge about the biology, and novice learners having low prior domain knowledge about the biology to assess learning with well-structured and ill-structured hypertext content. It was found that expert learners learned more from ill-structured content and novice learners learned more from well-structured hypertext content.

Another study (Al-Samarraie et al., 2013) on structured text representation, authors created a fixed template with seven segments for academic articles. The segments included title, introduction, problem statement, objective, method, analysis and result. They found that structured representation of text leads to high metacognition, attention, engagement, motivation among learners.

Therefore, novice learners who have less domain knowledge are benefitted from well-structured hypertext. Hence designing well-structured learning modules will benefit students who have low or high knowledge about the subject.

2.1.2 *Learners' Interest*

Interest has been categorized as a form of intrinsic motivation. Learner's interest could be content-specific and that too for specific subjects and tasks. The researchers have identified two types of interest – individual interest and situational interest. Individual interest is a long term affinity towards a specific subject area. Situational interest is a temporary emotional state due to situational stimuli. Situational interest could be due to the elements of the text, hypertext or it could be due to the content or domain being studied (Lawless et al., 2003).

The elements that activate situational interest among learners could be used in learning modules in order to make students read content.

2.1.3 *Complexity of Concept*

Concepts or tasks, to be learned, can be complex in nature. Domain like computer programming is considered a complex cognitive domain where learning and problem solving consume much of the cognitive resources imposing high cognitive load on programmers' cognitive system (Paas, n.d.; Robins, Rountree, & Rountree, 2003). In order to make students read content in the areas of complex domain, the complexity of the concepts or tasks must be simplified by applying cognitive models of learning.

One of the widely accepted cognitive models of learning in the field of education is Bloom's taxonomy. Bloom's taxonomy is the hierarchical model for higher order

learning in complex tasks. Bloom's taxonomy has been used in computer science domain as well (Johnson & Fuller, 2006; Thompson, Luxton-Reilly, Whalley, Hu, & Robbins, 2008). Therefore, applying Bloom's taxonomy in learning modules will make teaching complex concepts simpler.

2.2 Interactivity and Engagement

In section 2.1, one of the factors discussed that could lead readers to skip content is situational interest. Interest has been categorized as a form of intrinsic motivation. Motivation can be increased by engaging learners or readers with the system they are interacting with.

Learner engagement in e-learning has an impact on learning outcome (Ramesh, Goldwasser, Huang, Daume, & Getoor, 2014). Learner engagement is "the degree to which a learner feels involved or connected in a variety of educationally related activities" (Southerland & Nathaniel, 2010). Other authors define engagement as, "students' involvement in their own learning process" or "time or effort devoted by students to learn activities" (Rodriguez & Armellini, 2013). (O'Brien & Toms, 2008) describe learner engagement with the learning system, a 4 step process. It includes- 1) point of engagement, 2) period of engagement, 3) disengagement, and 4) reengagement.

The first step, point of engagement is when learners' engagement with the learning system begins. The engagement is influenced by aesthetics and informational content of the user interface which attract the users' attention and interest.

The second step, period of engagement, sustains the engagement initiated in step

1. Sustained engagement can be marked by users' focused attention and interest towards the task they are doing. Focused attention could be influenced by factors such as feedback generated by the system, the novel information and features of the interface. Users, in this step, also stayed engaged, first - due to the challenging task offered by the system, second – they think they are the in-charge or control the interaction with the system. Novelty is the sudden and unexpected change that occurs in an interface that catches users' attention and interest.

The third step, disengagement, is the when user decides to stop the activity because of his/her intrinsic decision or because of the external environment. This decision could be because of loss of interest, time constraint or external pressure or distractions. The fourth step, reengagement, when user decides to get back to the task which was stopped or left incomplete.

We summarize that engagement with the learning system begins due to the aesthetics of the user interface, interest of the learner towards the topic or content, novelty in the user interface, and motivation or specific goals of the learner. This engagement could be sustained for a longer period again due to aesthetics of the user interface, interactivity with the system, learners' feeling as the in-charge or controller of the interaction with the system, learner solving the challenging task, feedback provided by the system, and continuously evolving interest.

2.2.1 Interactivity

Engagement increases with interactivity. Interactivity with the learning environment motivates learners to learn (Moreno & Mayer, 2007). Interactivity in context of learning is the, “responsiveness to the learner’s actions during learning” (Moreno & Mayer, 2007). The types of interactivity in e-Learning environments include: *dialoguing, controlling, manipulating, searching* and *Navigating* (Moreno & Mayer, 2007). This research focuses on dialoguing and controlling types of interactivity, the remaining three are addressed by most web-based platforms. Dialoguing occurs when the learner answers questions and receives feedback to his/her input. Controlling means, the learner can determine or control the pace of the presentation.

2.2.1.1 Dialoguing - Assessment

Dialoguing help students to learn better, through the feedback provided by the learning environment which reduces extraneous cognitive load in the working memory (Hessler & Henderson, 2013; Moreno & Mayer, 2007). Dialoguing in e-learning environments has been implemented primarily through assessments. Assessments are conducted to test student learning after they have gone through the content. There are two types of assessments: formative and summative.

Summative assessments provide the judgment about the student achievement at the end of the course or instruction (Reeves, 2000). Formative assessment is aimed to evaluate, assist and promote student learning by providing continuous feedback about the topic learned during the period of instruction (Iahad, Dafoulas, Kalaitzakis, & Macaulay, 2004; T. H. Wang, 2007). Evaluating student learning through assessments include

various formats including multiple-choice questions, true-false, fill-in-the-blank, short answer and essays (Kuechler & Simkin, 2003).

In Multiple Choice Questions and True or false, students have to select a correct response from the list of pre-written responses. In short-answer, fill-in-the-blank, and essay type questions, alternatively called as constructed responses, students have to construct the answers on their own, based on their understanding of the topic they have learned (Kuechler & Simkin, 2003).

There have been extensive arguments about which type of test format will help students to learn better. Research suggests that multiple choice type questions are regarded as the most valuable and applicable form of the test to measure learning objectives such as “inferential reasoning, reasoned understanding, sound judgment and discrimination” (Iz & Fok, 2007), hence, infusing deep learning in a student.

Multiple choice questions sometimes have been regarded as infusing surface level or rote learning among students (Iz & Fok, 2007; Scouller, 2006). Rote or surface level learning involves recalling of factual knowledge. While constructed responses are considered to infuse deeper learning, they are highly subjective and difficult to measure on e-learning systems, and therefore are less popular among educators (Iz & Fok, 2007; Kuechler & Simkin, 2003).

2.2.1.2 *Dialoguing - Feedback*

Feedback to student test answers has been considered beneficial in student learning (Thalheimer, 2008). (Iahad et al., 2004) suggests that rich feedback is one of the requirements of learner centered environment. Feedback, depending upon the response-

time and amount, in e-learning environments, are of the following types, 1) time: immediate and delayed feedback; 2) amount: Knowledge of Results (KR), Knowledge of Correct Response (KCR) and Elaborate Feedback (EF)(Stuart, 2004; Thalheimer, 2008; van der Kleij, Eggen, Timmers, & Veldkamp, 2012).

Immediate feedback is, when a student receives feedback immediately upon submit whereas feedback given sometime after the submission of the response is called delayed feedback. : Knowledge of Results (KR) type of feedback tells only whether answer is correct or incorrect, Knowledge of Correct Response (KCR) tells whether answer is correct or incorrect along with the correct answer, Elaborate Feedback (EF) tells whether answer is correct or incorrect along with concise explanation of the correct answer (van der Kleij et al., 2012).

Several research studies have suggested that there is an impact of the amount of feedback presented, on the student learning. The Knowledge of Correct Response (KCR) and Elaborate Feedback (EF) types of feedback have more impact on student learning than KR type of feedback. A few studies have suggested that immediate feedback improves student learning and is better than delayed feedback in most of the circumstances because when student receives feedback immediately, student can relate it to current learning as opposed to delayed feedback which could be given to learner after hours or days from the time of the test taken (Azevedo & Bernard, 1994; Stuart, 2004; van der Kleij et al., 2012).

Another type of the feedback that is seen in various assessments is answer-until-correct. Not much research has been conducted in this area but a few researchers have found this methodology beneficial. The report by (Thalheimer, 2008) concludes that this

type of methodology would benefit for those learners who have learned materials well, as compared to those who have not.

2.2.1.3 Controlling the Presentation

Controlling means, the learner can determine or control the pace of the presentation. Controlling help students learn better by allowing them to control the pace of the presentation. Controlling reduces the extraneous load by allowing students to process smaller chunks of information in the working memory at their own pace.

2.2.2 Approaches to Learning – Surface & Deep

Students can take different learning approaches to learn content. Taking these learning approaches, student can learn content in depth or on the surface. These approach are called deep learning and surface learning respectively (Entwistle, 2000). In deep learning, students generate intrinsic motivation and interest in the learning content. They aim at understanding the meaning of the learning material by relating different parts of the concepts and come up with new ideas based on their prior knowledge. They learn the content in such a way that they are able to apply it in the real world (Chin & Brown, 2000). In surface learning, learners do not put effort and are not involved in learning. They attain the factual or rote knowledge without any understanding of the concept (Entwistle, 2000; Floyd, Harrington, & Santiago, 2009).

Research indicates that student engagement with the learning content infuses deeper learning. It is the motivation and interest for the content which inculcates learners to adopt the deep learning approach (Chin & Brown, 2000; Entwistle, 2000).

In summary, it is desirable for learning modules to have elements that invoke

situational interests among learners. Situational interest is an intrinsic motivation that can be increased by engaging learners with the content using interactivity. Interactivity in learning modules could be incorporated through dialoguing and controlling.

2.3 Learning in Interactive versus non-interactive systems

Any kind of educational intervention has a positive effect on student learning (Hattie, 2013). Then, do interactive and non-interactive systems with similar content will have the same learning effect?

A similar study conducted by Evans and Gibbons (Evans & Gibbons, 2007) showed that both interactive and non-interactive systems with same content have same learning when assessment questions examine retention or recall. But, interactive systems performed significantly better than non-interactive system when assessment involved problem-solving.

Another study by Wang et. al. (P.-Y. Wang, Vaughn, & Liu, 2011) examined impact of animation interactivity on novices' learning of introductory statistics. The study comprised of three groups – 1) static group – provided with static material, 2) simple animation group – animation with input manipulation, and 3) practice group – animation with practice and feedback. The results showed animation interactivity significantly improved students' understanding and lower level applying.

A study by Mayer et. al. (Mayer, Dow, & Mayer, n.d.) found that students performed significantly better on problem solving transfer test due to the interactive feedback provided by the system compared to non-interactive version.

In summary, the interactive and non-interactive learning systems will show same

student learning for the questions that assess retention of knowledge. The interactive learning systems will show significantly higher student learning as compared to non-interactive systems for the questions where knowledge is to be applied.

2.4 Eye tracking and Reading Research

This research uses eye-tracking as a method to measure content skipping.

Ongoing research in various domains using eye-tracking is based on Eye-Mind link theory. The theory states that there is a link between human mind and the eyes i.e. eyes move in parallel with the mind (Holmqvist, Holsanova, Barthelson, & Lundqvist, 2003). Therefore, measuring eye-movements of a person looking at an object reveal various characteristics of the person and how do they perceive these objects. The measuring of eye-movements is called eye-tracking and the device used to measure is called eye-tracker.

Eye-tracking has been extensively used in the research areas involving information processing such as reading, scene perception, visual searching, music reading, and typing (Rayner, 1998). Eye-tracking has also been applied to investigate human computer interactions including usability of systems, content presentation formats etc. (Ariasi & Mason, 2010; Atterer, Wnuk, & Schmidt, 2006; Chuang & Liu, 2011; Sharmin, Špakov, & Rähä, 2012). More recently, eye-tracking is used in computing education research to determine computer programmers' reading and understanding of computer programs (Bednarik, Busjahn, Schulte, & Tamm, n.d.; Busjahn et al., 2014, 2015).

Several eye-tracking tools are currently available in the commercial market

including trackers from SR Research, Sensomotoric Instruments (SMI) and Tobii (<http://www.tobii.com>) (Holmqvist et al., 2011).

There are two major characteristics of eye movements – 1) fixation and, 2) saccades. Fixation is the settling of the eye gaze on stimuli for a minimum period of time. Saccade is a quick movement of the eyes from one fixation to another. Eye movements in reading consist of a series of fixations and saccades. In English reading, the saccades could be forward (left to right direction) or backward (right to left direction, also called regressions) (Rayner, 1998).

Reading is a well-defined movement of the eye from left to right, with approximately one stop at each word and small jumps (saccades) between them (Holmqvist & Wartenberg, 2005). For skilled readers fixations last about 200-250 ms and forward saccades have amplitudes of 7-9 letters. Most (about 80 percent) of the saccades are forward and about 10-15% are regressions. Regressions could be – only few letters long (for efficient reading after a long saccade), short-within word (due to problems in processing current fixated word) or too long (more than 10 letter spaces, due to non-understanding of text) (Rayner, 1998).

During reading, majority of the words in a text are fixated while many are skipped. Among these, content words are fixated about 85% times and functional words about 35% (as functional words tend to be short). Also, the probability of fixating on a word increases as word length increases (Rayner, 1998). To measure reading, several reading detection algorithms have been implemented (Holmqvist et al., 2011). The simplest reading detectors use saccadic amplitudes to detect reading. Saccades are short when reading in horizontal direction with different fixation duration and long when

scanning (Holmqvist et al., 2011).

(Campbell & Maglio, 2001) developed an algorithm to detect online reading using three saccadic criteria – distance (long versus short, using pixels information), direction (right, left, up, down) and axis (x versus y). The algorithm 1) quantizes eye movements from eye-tracker's raw data by averaging every three data points; and 2) detects evidence of reading. To detect the reading evidence - each saccade is assigned a score (positive when eye moves to right and negative when eye moves to left); the scores are summed and compared to a threshold score. The authors reported high accuracy.

Using modification to (Campbell & Maglio, 2001) algorithm, (Buscher, Dengel, & van Elst, 2008) developed reading-skimming detector. The algorithm includes following steps- 1) detection of fixations, 2) feature detection based on classification of transitions from one fixation to another (Refer **Figure 2**), 3) accumulation of scores associated with the feature and 4) determining if scores exceed reading and skimming thresholds. The feature is detected based on transition's distance (short, long in letter spaces) and direction (forward and regressions) from one fixation to another. Using letter spaces is considered appropriate metric to measure saccadic amplitude in reading (Rayner, 1998).

Distance and direction in letter spaces	Feature	Reading detector score s_r	Skimming detector score s_s
$0 < x \leq 11$	Read forward	10	5
$11 < x \leq 21$	Skim forward	5	10
$21 < x \leq 30$	Long skim jump	-5	8
$-6 \leq x < 0$	Short regression	-8	-8
$-16 \leq x < -6$	Long regression	-5	-3

Figure 2: Reading-skimming detection algorithm

Another algorithm by (Simola, Salojärvi, & Kojo, 2008) developed hidden Markov model with an accuracy of 60% to detect three different reading tasks including – simple word search; finding a sentence that answers a question; and, choosing a title from the list of titles. Several studies have used other metrics including - fixation duration and reading depth to detect the portions of text read. Fixation during reflects the time to process reading text; and reading depth, also called as reading ratio, reflects proportion of an area looked at (Holmqvist et al., 2011).

In summary, students’ content skipping can be measured using eye-tracking. Using eye-movements (fixation, saccades), two metrics-1) reading scores (using a reading detection algorithm in eye-tracking by Buscher et al. 2008 (Buscher et al., 2008)) and 2) reading depth (number of words fixated in a given area of text) can be used to detect whether how much content has been read.

2.5 Usability and Learning

In e learning, usability is significantly important as it influences students’ learning (Meiselwitz & Sadara, 2008). Usability is “The extent to which a product can be used by

specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (Ssemugabi & de Villiers, 2007). If an e-learning application is not usable enough, learner’s would not spend more time in learning the content (Ardito et al., 2005). In order for e-learning applications to be usable, the applications should- 1) be interactive and provide feedback, 2) motivate 3) provide suitable tools, and 4) avoid distractions, to learners (Ardito et al., 2005). More importantly, in addition to these- the interface, content, tools and tasks in the e-learning application should support pedagogical objectives, apart from being attractive and engaging (Ardito et al., 2005).

Usability evaluation is a method to assess or improve the applications by identifying problems and suggesting improvements (Ssemugabi & de Villiers, 2007). Several usability evaluation methods exist including analytical, expert heuristic evaluation, questionnaire, observational, and experimental methods (Brinck, Gergle, & Wood, 2001; Shneiderman, Plaisant, Cohen, & Jacobs, 2010). Several factors including efficiency, time, cost-effectiveness, ease of application, and expertise of evaluators determine which evaluation method to select. Usability evaluation using questionnaire is a popular method due to the following advantages -1) can collect a large amount of data and, 2) it is usually quick and cost-effective to administer and to score (Zaharias & Poylymenakou, 2009).

Several e-learning usability questionnaires exist. Usability questionnaire for e-learning systems contain web and instructional design attributes in addition to general system usability attributes (effectiveness, efficiency and satisfaction). Zaharias and Poylymenakou 2009 in (Zaharias & Poylymenakou, 2009) developed a usability questionnaire that measured attributes related to content, learning & support, visual design, navigation, accessibility, interactivity, self-assessment & learnability and motivation to

learn. Ssemugabi and de Villiers 2007 in (Ssemugabi & de Villiers, 2007) assessed usability using three criteria -1) interface usability, 2) educational website-specific criteria, and 3) learner-centered instructional design. Ardito et al. 2005 in (Ardito et al., 2005) used presentation, hypermediality, user activity and application proactivity dimensions to assess usability in e learning.

Usability studies that compare two learning systems with same content and different designs have shown significantly higher student performance for usable systems. E.g. Avouris et al. 2001 in (Avouris, Dimitracopoulou, Daskalaki, & Tselios, 2001) compared “student testing and self –assessment” module in two different learning environments – “WebCT” and “IDLE”. WebCT were found to be significantly usable than IDLE and students’ performance score was significantly better in WebCT than IDLE. Another study by Parlangeli et al. 2011 in (Parlangeli, Mengoni, & Guidi, 2011) compared a usable and a non-usable teaching website with same content followed by learning assessment. They found, non-usable system negatively affected the learning process among students.

In summary, – 1) usability evaluation is an important component in e learning in order to assess or identify issues in the system; 2) Using questionnaire is an appropriate evaluation method as it is easy to collect large amount of data, administer and score; and 3) usable systems have shown improved student learning.

2.6 Theoretical Framework

In summary, literature review in section 2.1 attributes content skipping in a large web-based hypertext pages to – 1) scrolling up and below the document, 2) flexibility to click any of the hyperlinks that links within or outside the page to gain knowledge and

thus losing context of the original text, 3) reading strategy adopted by readers which determines what to read and what to skim, 4) perception of the large length of document among readers, 5) lack of prior or domain knowledge about the content in a reader, 6) content-structure (ill-structured content may lead less domain knowledge learners to skip content), 7) complexity of content (Highly complex content may lead to content skipping), and 8) lack of situational interest.

Section 2.2 suggests, it is desirable for learning modules to have elements that invoke situational interests among learners. Situational interest is an intrinsic motivation that can be increased by engaging learners with the content using interactivity. Interactivity in learning modules could be incorporated through dialoguing and controlling.

Section 2.3 suggests, the interactive and non-interactive learning systems will show same student learning for the questions that assess retention of knowledge. The interactive learning systems will show significantly higher student learning as compared to non-interactive systems for the questions where knowledge is to be applied.

To overcome the issues in linear modules, we proposed to incorporate e-learning design principles of segmentation and interactivity (Raina, Taylor, & Kaza, 2015). Segmentation implies breaking large content into smaller chunks and present one chunk at a time on a single screen. Segmentation makes processing, retention and recalling of information easier (Clark & Mayer, 2011; Moreno & Mayer, 2007). In addition, to increase engagement with the module interface, research suggests increasing user-system interactivity (Quinn, 2005). Interactivity in e-learning is the “responsiveness to the learner’s actions during learning.” We proposed to increase interactivity with dialoguing and controlling. The process of a learner answering a question and receiving feedback on

his/her input is referred to as dialoguing. Dialoguing improves learning (Thalheimer, 2008), as learners can relate feedback to the current content. Controlling implies that the learner can determine the pace of the presentation. Controlling helps students learn better by allowing them to process information at their own pace.

Overall, segmentation breaks large content into smaller chunks and presents them one at a time, which may result in less reading and less skipping of content (Protopsaltis & Bouk, 2005; Tseng, 2008) . Less skipping of content may lead to increased learning (Rudestam & Schoenholtz-Read, 2010). Interactivity (dialoging and controlling) on segmented chunks leads to engagement and enforces learning (Zhang, 2010)

3. Method

This research included 1) enhancement of original linear learning modules to incorporate e-learning design principles of segmentation and interactivity using security injections @Towson cybersecurity modules, 2) testing the effectiveness of enhanced modules through four quasi-experimental studies, each addressing the research questions described in introduction chapter.

This chapter presents the research questions and describes the method used including sample, procedure, instruments, hypotheses and the data collection and analysis for each research question.

3.1 Research Questions

RQ1: Can the use of learning modules with segmentation reduce content skipping as compared to linear modules?

This research examines content skipping based on reading scores and reading depth. Higher reading scores and depth means more coverage and less skipping (Biedert, Hees, Dengel, & Buscher, 2012; Buscher et al., 2008; Holmqvist et al., 2011). Therefore, RQ1 is answered using RQ1a and RQ1b.

RQ1a: Can the use of learning modules with segmentation show significantly higher reading scores as compared to linear modules?

RQ1b: Can the use of learning modules with segmentation show significantly higher reading depth as compared to linear modules?

RQ2: Can the use of learning modules with segmentation and interactivity

increase student engagement as compared to linear modules?

RQ3: Can the use of modules with segmentation and interactivity increase student learning as compared to linear modules?

Previous research has shown that while retention of knowledge is not affected by the interactive nature of the system (Evans & Gibbons, 2007; Hattie, 2013; Mayer et al., n.d.; P.-Y. Wang et al., 2011), students perform significantly better on application of knowledge using interactive systems (Evans & Gibbons, 2007; P.-Y. Wang et al., 2011). Based on these findings, RQ3 is answered using RQ3a and RQ3b.

RQ3a: Can the use of learning modules with segmentation and interactivity show same learning as compared to linear modules for questions that assess retention of knowledge?

RQ3b: Can the use of learning modules with segmentation and interactivity show significantly higher learning as compared to linear modules for questions that assess applying of knowledge?

RQ4: Are learning modules with segmentation and interactivity significantly more usable than linear modules?

3.2 Research Design

RQ1 was tested using an experimental control-group treatment-group design. The independent variables were 1) linear module (control) and 2) segmented-interactive Module (treatment). The dependent variables were 1) reading scores and 2) reading depth.

RQ2 was tested using a quasi-experimental post-survey only control-group

treatment group design. The independent variables were 1) linear module (control) and 2) segmented-interactive module (treatment). The dependent variables were engagement scores.

RQ3 was tested using a quasi-experimental pre-survey post-survey control-group treatment-group design. The independent variables were 1) linear module (control) and 2) segmented-interactive module (treatment). The dependent variables were 1) general security-awareness scores, 2) secure-coding awareness scores, 3) ability to apply secure-coding knowledge scores, 4) phishing awareness and 5) ability to apply phishing knowledge.

RQ4 was tested using a quasi-experimental post-survey only control-group treatment group design. The independent variables were 1) linear module (control) and 2) segmented-interactive module (treatment). The dependent variables were overall usability scores.

3.3 Sample

Towson University is a mid-size institution with over 18,000 undergraduates. The Computer and Information Sciences Department comprises two majors, Computer Science (CS), Computer Information Systems (CIS) and Information Technology (IT). Currently, there are approximately 300 students in each major.

The department offers core programming courses including CS0 (using C++) and a computer literacy course for all majors and non-majors. Each of the courses includes a lecture and a lab component. The courses are 15 week four-credit (CS0) and three-credit

(computer literacy) classes and are described below:

CS0: *COSC175 - General Computer Science* logic course taught in C++.

Computer Literacy: *COSC111 – Information & Technology for Business*.

RQ1 was tested using a random sample (randomly drawing chits that labels either control or treatment groups) of students from two sections of CS0 course in fall 2015 and spring 2016. Both the sections were taught by the same instructor. A total of 30 (15 in treatment and 15 in control) students participated in the study.

RQ2 was tested using a convenience sample of students from four sections of CS0 course in fall 2014. Two sections were taught by the same instructor and other two by different instructors. Instructor teaching two sections was requested to assign one section to control and other section to treatment group. One section from other two instructors was assigned to control group and another to treatment. A total of 116 (60 students in treatment and 56 in control) students participated.

RQ3 was tested using a convenience sample of students from two sections of CS0 and three sections of computer literacy course in spring 2015. In CS0, both sections were taught by the same instructor, one section was a control group and other a treatment group, and a total of 53 (26 in treatment and 27 in control) students participated. In computer literacy, two sections were taught by the same instructor where one section was control and another treatment; third section taught by a different instructor was divided into two groups (control and treatment). A total of 94 (48 in treatment and 46 in control) students participated in this study.

RQ4 was tested using a convenience sample of students from six sections of CS0 and two sections of CS1 in fall 2015. Three sections in CS0 and one section in CS1 were assigned to control group and other sections in CS0 and CS1 to treatment. A total of 538 (332 in treatment and 206 in control) students participated.

3.4 Instruments

Instruments used in this research are six surveys- 1) Pre-survey and 2) post-survey to assess software security learning; 3) Pre-survey and 4) Post-survey to assess general security learning; 5) engagement survey to assess student engagement, and 6) usability survey to assess overall module usability; and an eye-tracking device to examine students' content skipping. Each instrument is described below.

3.4.1 Pre and Post Software Security Survey

The survey instruments (both pre-survey and post-survey) were used to test RQ3 using CS0. The instruments were derived from previous security injections studies (B. Taylor & Kaza, 2011b). Both the pre-survey and post-survey include multiple choice questions related to student demographics, secure coding awareness and general software security awareness.

Secure coding awareness include 5 questions - 2 integer overflow, 2 input validation and 1 buffer overflow; and, 4 questions for general software security awareness. In addition, 3 code segments were added to the post-survey to assess students' ability to apply secure coding knowledge. The code segments were developed by the senior instructors teaching CS0. The students were to identify the potential security

vulnerability in the code segments. See **Table 3** for security awareness questions and **Table 2** for code segments (ability to apply).

Identify the potential security issues in the following code segment: (Check all that apply)
<p>Code Segment 1</p> <pre>float price; float totalPrice; cout << "Enter Price" << endl; cin >> price; totalPrice = price + price*.06;</pre> <p>Code Segment 2</p> <pre>//assume i < INT_MAX and j < INT_MAX int calc (int i, int j)) { int result = i * j; return result; }</pre> <p>Code Segment 3</p> <pre>//assume n < INT_MAX void input(float temperatures[], int n) { for (int i = 0; i < n; i = i + 1) { cout << temperatures[i] << endl; } }</pre>

Table 2: Code Segments (Ability to apply)

Secure-coding Awareness
Integer Overflow occurs ..
Integer Overflow is caused by ..
Invalid input can come from ..
Which of the following should your well designed program do before processing user input?
Which programming mistake is one of the major vulnerabilities in today applications?
Software-security Awareness
What are the possible consequences of insufficient computer security?
Security Software and Software Security are the same:
When developing secure systems, where does security fit in?
Software security vulnerabilities are the result of software bugs and flaws:
Your code is completely secure if:

Table 3: Survey questions (Software Security Awareness)

3.4.2 Pre and Post Computer Literacy Survey

The survey instruments (both pre-survey and post-survey) were used to test RQ3 using computer literacy. The instruments were derived from previous security injections studies (Turner, Taylor, & Kaza, 2011). Both the pre-survey and post-survey include multiple choice questions related to student demographics, computer security including phishing.

Phishing awareness include 4 questions. In addition, a sample email was presented to in order assess ability to identify phishing. See **Table 4** for phishing awareness questions and **Table 5** for sample e-mail to identify phishing (ability to apply).

Phishing Awareness
Phishing is
The following are characteristics of suspicious email...
Never give out personal information upon an email request..
Consider the following email: Is this email legitimate?

Table 4: Survey Questions (Phishing Awareness)

Ability to apply Phishing Knowledge
Consider the following email:
<p>From: Help Desk <online2793774@telkomsa.net> Date: June 20, 2014 at 7:57:55 AM PDT To: info@cs.stanford.edu Subject: update</p> <p>It had been detected that your cs-stanford-edu email account. Mail delivery system had been affected with virus. Your email account had been sending virus included with your mail to recipient's account and as such a threat to our database. You'll need to update the settings on your cs-stanford-edu email account by clicking on this link: http://forms.logiforms.com/formdata/user_forms/66949_9366478/321793</p> <p>From CS. Standford ITS Helpdesk</p>
Is the above email, legitimate or fraudulent ?
What makes you decide, the above email is legitimate or fraudulent ? Discuss elaborately.

Table 5: Sample e-mail to identify Phishing (Ability to apply)

3.4.3 Student Engagement Survey

The instrument tests RQ2. The survey instrument used in this study measured student demographics and student engagement. While questions related to student demographics were derived from previous security injection studies (B. Taylor & Kaza, 2011b), which measured students' gender, age-group, ethnicity and major, we adapted a set of eight item questions from a well-tested User Engagement Scale (UES) (O'Brien & Toms, 2010) to measure student engagement.

The eight item student engagement questions were recorded from 1 to 5 on a five point Likert scale 1 representing 'strongly disagree' and 5 representing 'strongly agree'. (See **Table 6** for sample survey questions.). A reliability test was conducted to test the internal consistency of the survey. The cronbach's alpha, for eight-item engagement questions, was found to be 0.74, which suggested good internal consistency. The survey was administered online on student voice.

	Student engagement
Q1	I felt deeply engrossed while completing security injection modules using this web-based platform.
Q2	I get so involved while completing security injection modules using this web-based platform that I forget everything.
Q3	While completing the security injection modules using this web-based platform, I tend to block out conversations with others around me.
Q4	The Security Injection modules presented on this platform hold my attention.
Q5	Using this web-based platform excited my curiosity to learn cybersecurity principles.
Q6	Time seemed to go by very quickly when I use this web-based platform for completing Security Injection module.
Q7	The screen layout of this web-based platform for Security Injection modules was visually pleasing.
Q8	Using this web-based platform for completing Security Injection modules was attractive.

Table 6: Student engagement survey questions

3.4.4 Module Usability Survey

The survey instrument was used to test RQ4. The survey measured student demographics and usability. While questions related to student demographics were derived from previous security injection studies (B. Taylor & Kaza, 2011b), which measured students' gender, age-group, ethnicity and major, we adapted a set of eighteen item questions from three different e-learning usability questionnaires to measure usability in following categories - learnability; navigation; accessibility; consistency; visual design; interactivity; instructional assessment; instructional feedback; learning guidance & support; efficiency; effectiveness; and user satisfaction.

The eighteen item usability questions were recorded from 1 to 5 on a five point Likert scale 1 representing 'strongly disagree' and 5 representing 'strongly agree' (See **Table 7**). A reliability test was conducted to test the internal consistency of the survey. The cronbach's alpha, for eight-item engagement questions, was found to be 0.97, which suggested strong internal consistency. The survey was administered online on student voice platform.

3.4.5 Eye-tracking Apparatus

An eye-tracking apparatus was used to test RQ1. The eye movements of each participant were recorded using a tobii T60 eye tracker with tobii studio 3.0 software package. The eye-tracker was installed on a windows 7 operating system with 64 GB memory, 3 GHz processor and 1 TB hard drive. The device was placed on the bottom frame of a 17 inch LCD monitor (see **Figure 3**) with a resolution of 1280 X 1024 pixels

and frequency 60 Hz. The eye fixations were detected using tobii's I-VT filter fixation detection algorithm. A second monitor, connected to the eye-tracking computer and kept at a distance in the same room, was used to monitor participants' eye-track status.



Figure 3: Eye-tracker mounted on a 17" inch monitor

Usability Questionnaire		Adapted from
Learnability		
1. Instructions to use the module were clear.		(Hegarty, 2005)
Navigation		
2. I found it easy to navigate around the module.		(Teoh & Neo, 2007)
Accessibility		
3. The module is easy to launch.		(Zaharias & Poylymenakou, 2009)
Consistency		
4. The fonts, colors, and sizes are consistent throughout the module.		(Zaharias & Poylymenakou, 2009)
5. The module maintains an appropriate level of consistency in its design from one part/section of the module to another.		
Visual Design		
6. I found the interface clear, structured and appealing.		(Teoh & Neo, 2007)
7. Text and graphics are legible.		(Zaharias & Poylymenakou, 2009)
8. Fonts (style, color, saturation) are easy to read.		
Interactivity		
9. The module does not provide too many long sections of text to read without meaningful interactions.		(Zaharias & Poylymenakou, 2009)
10. The module engaged me in interactive tasks that are closely aligned with the learning goals and objectives.		
11. The module used interactive activities to gain the attention, sustain the interest, and maintain my motivation.		
Instructional Assessment		
12. Questions in the module enhanced my understanding of cybersecurity ideas and concepts.		(Zaharias & Poylymenakou, 2009)
13. Security checklist in the module enhanced my understanding of cybersecurity ideas and concepts.		
Instructional Feedback		
14. Feedback on activities is clear and helpful in learning.		(Zaharias & Poylymenakou, 2009)
Learning Guidance & Support		
15. The module provides guidance and support to complete individual sections including learning activities.		(Zaharias & Poylymenakou, 2009)
Efficiency		
16. I was able to complete the module quickly.		(Lewis, 1995)
Effectiveness		
17. I was able to effectively complete the module.		(Zaharias & Poylymenakou, 2009)
Satisfaction		
18. I was satisfied with the module.		(Lewis, 1995)

Table 7: Usability Questionnaire

3.5 Learning Modules

This research uses security injection@Towson learning modules to examine the research questions. These learning modules target key secure coding concepts including integer error, buffer overflow, and input validation in various programming languages, for Computer Science 0 (CS0), Computer Science 1 (CS1), and Computer Science 2 (CS2); and general security concepts, such as phishing, passwords, and cryptography for use in Computer Literacy courses.

The control group uses original security injection modules and the treatment group uses enhanced modules. The original modules are linear and presented on a single webpage.

The original module begins with a background section to describe the problem with examples, followed by a “Code Responsibly” section (includes methods to avoid security issues), a laboratory assignment with a security checklist, and discussion questions. Students submit the laboratory assignment and discussion question answers to their instructors as a text document to receive the grades and feedback.

The enhanced modules are segmented and interactive with same content as original. In enhanced modules, original content is broken per section (background, code responsibly, laboratory assignment, discussion questions) and each section is presented, one at a time, on the screen. In this fashion, the reader views a small amount of content at a time. Each section in a module is auto-graded using built-in functionality for text and multiple-choice questions. In the background and code responsibly sections, students are required to go through the content and answer a set of checkpoint questions. Each

question provides immediate feedback on submit. The student cannot advance to the next section until all questions are answered correctly.

3.6 Procedure

To examine RQ1, the control group completed integer error module using linear format and the treatment group completed using segmented-interactive format. Each participant was allocated different time slots (one hour each) due to availability of a single eye-tracking device. For each participant, the experiment involved four steps -1) eye calibration; 2) administering demographics survey 3) completing the module 4) administering the usability survey.

Each participant showed up in their allocated one hour time slot in the human computer interactions laboratory. Participants were given brief introduction about the experiment with the following description –

“Today you will be learning about a major software security vulnerability (Integer Overflow) using Security Injections @Towson cybersecurity module. There are two versions of these modules – 1) linear and 2) segmented. You will be asked to draw a random chit, that labels the version of the module you will be completing, from a box of chits. Depending on the module version, you will open a select document that contains instructions to complete the tasks where first, you will complete a short demographics survey; second, you will complete the module; and third, you will complete the usability survey. During the experiment your eye-movements will be recorded using an eye-tracker for data collection purposes only. Your identity will be kept completely secured. To capture eye-movement, your eyes will be first calibrated. If you have any questions

during the experiment, raise your hands to indicate. ”.

The calibration includes three steps process – 1) eye detection, 2) calibration, and 3) result acceptance (Refer Fig. 5.). In eye-detection, participants were asked to sit on a chair in a comfortable position in front of the eye-tracker and look at the monitor. The participant's positions were adjusted until eyes were detected at the center of eye-track status window to be able to capture eye-movements accurately with high precision. The allowable distance of the participants' position from the monitor was 50 cm – 80 cm. In calibration, participants were asked to look at the center point of a moving ball on a 9 point calibration view. In result acceptance, the calibration results are presented with an option to accept the calibration or re-calibrate. The calibration was accepted only when green dots were within each 9 point circles otherwise re-calibration was performed. After calibration, participants took demographics survey, completed integer error module and usability survey in sequence.

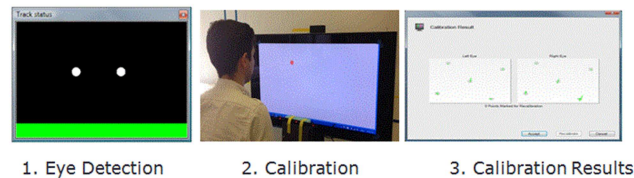


Figure 4: Calibration process steps

To examine RQ2, the study was conducted during the laboratory sessions which were at different times for each section. Three modules - integer error, input validation and buffer overflow - were introduced, in that order, with approximately four weeks between the interventions. Both control and treatment groups were administered a

student engagement survey at the end of the semester.

To examine RQ3, in CS0 course, the study was conducted during the laboratory sessions, which were at different times for each section (11-11:50 AM (control), 12-12:50 PM (treatment)). Three modules - integer error, input validation and buffer overflow - were introduced, in that order, with approximately four weeks between the interventions. Both groups were administered a pre-survey at the beginning and a post-survey at the end of the semester.

In computer literacy course, the study was conducted during the laboratory sessions, also at different times for each section (two sections taught by same instructor: 11-11:50 AM (control), 12-12:50 PM (treatment); third section taught by a different instructor (10 – 11:15 AM (students divided into two groups, control and treatment))). A module in phishing was introduced during the eighth week of the semester. Both groups were administered a pre-survey at the beginning and a post-survey at the end of the module to assess general security awareness retention. In addition, a sample of phishing email was given to participants, after the post-survey, to identify any sentence, phrase or word that makes the e-mail a suspected phish, to assess student's ability to apply security knowledge.

To examine RQ4, the study was conducted during the laboratory sessions, which were at different times for each section. Three modules - integer error, input validation and buffer overflow - were introduced, in that order, with approximately four weeks between the interventions. Both groups were administered a usability survey at the end of each module.

3.7 Hypotheses

To examine RQ1, based on the reading scores and reading depth we propose the following hypothesis:

H1: The mean reading scores in treatment group will be significantly higher than the mean score for control group.

Rationale – Due to less content in segmented modules, students will read more and skip less as compared to linear modules where there is large amount of content on a single page. This links to RQ1a.

H2: The mean reading depth in treatment group will be significantly higher than the mean reading depth in control group.

Rationale – In segmented modules, readers will fixate on more words as compared to linear modules. This links to RQ1b.

To examine RQ2, Based on the survey scores, we proposed the following hypothesis to compare Security Injections 1.0 and Security Injections 2.0 (treatment group) on the following dependent variable: student engagement score.

H2: The mean of survey scores for student engagement in the treatment group will be significantly higher than the mean of the survey scores for student engagement in the control group.

To examine RQ3, based on the pre-survey and the post-survey scores, we proposed the following set of hypotheses to compare Security Injections 1.0 (control group) and Security Injections 2.0 (treatment group) on the following dependent variables: secure coding awareness, general software security awareness, and ability to

apply secure coding knowledge.

H3a: The post-survey scores for secure coding awareness (measuring retention) will be significantly higher than the pre-survey scores for secure coding awareness in both control and treatment groups.

Rationale – As discussed in section 2.2, any kind of educational intervention has a positive effect on student achievement. In addition, interactive and non- interactive systems with same content will have the same learning on recall or retention assessments.

H3b: The post-survey scores for secure coding awareness (measuring retention) for the treatment and the control group will not be significantly different.

Rationale – As discussed in section 2.2, both interactive and non- interactive systems with same content will have the same learning on recall or retention assessments. Secure coding awareness assesses retention of integer overflow, input validation and buffer overflow knowledge.

H3c: The post-survey scores for general software security awareness (measuring retention) will be significantly higher than the pre-survey scores for general software security awareness in both control and treatment group.

Rationale – Same as H3a

H3d: The post-survey scores for general software security awareness (measuring retention) for the treatment and the control group will not be significantly different.

Rationale – As discussed in section 2.2, both interactive and non- interactive systems with the same content will have same learning for recall or retention assessments. General software security awareness assesses retention of general software security knowledge.

H3e: The post-survey scores for phishing awareness (measuring retention) will be significantly higher than the pre-survey scores for phishing awareness in both control and treatment groups.

Rationale – As discussed in section 2.2, any kind of educational intervention has a positive effect on student achievement. In addition, interactive and non- interactive systems with same content will have the same learning on recall or retention assessments.

H3f: The scores for ability to apply secure coding knowledge in the treatment group will be significantly higher than the control group.

Rationale- As discussed in section 2.2, interactive systems will show significantly higher learning on problem-solving (ability to apply) assessments than the non-interactive systems with same content. The students apply their secure coding knowledge to identify security vulnerability in the code segments.

H3g: The scores for ability to apply phishing knowledge in the treatment group will be significantly higher than the control group.

Rationale- As discussed in section 2.2, interactive systems will show significantly higher learning on problem-solving (ability to apply) assessments than the non-interactive systems with same content. The students apply their secure coding knowledge to identify security vulnerability in the code segments.

H3a– H3e addresses RQ3a and, *H3f – H3g* addresses RQ3b

To examine RQ4, based on the survey scores, we proposed the following hypothesis to compare linear modules (control) and enhanced modules (treatment).

H4: The mean of survey scores for overall usability in the treatment group will be

significantly higher than the mean of the survey scores for overall usability in the control group.

3.8 Data Analysis

In RQ1, in order to compare students' reading between linear and segmented modules, reading detection algorithm by Buscher et al. 2008 (Buscher et al., 2008) and reading depth was used. We picked this algorithm because of the following reason- 1) The algorithm uses fixation points (unlike raw x-y coordinates by (Campbell & Maglio, 2001)) to detect saccades, and we use Tobii eye-tracking software that can generate fixation points for eye-movements; 2) The algorithm uses letter spaces to detect saccadic amplitudes which is considered most appropriate metric; 3) Implementing the algorithm is easier and time efficient. The raw data was processed from eye-tracker that involved following steps (See Figure 5):

1. *Data Export* - The eye movement data from the eye-tracker was exported for each participant in .tsv format using tobii studio 3.0.
2. *Filter Reading Data* – Each participants' reading data was extracted from .tsv file using start and end reading times per line. The timings were manually taken from the recorded videos. The reading data was extracted per line to exclude irrelevant eye movements due to page scrolling.

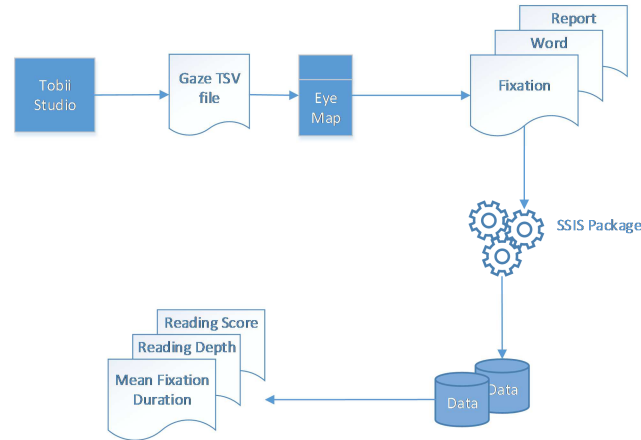


Figure 5: Data processing in Eye-tracking

3. *Aligning Fixations to Text Line* – Due to hardware inaccuracies in eye-tracking device, there are chances when fixation points don't show on the text while participant is reading. The fixation points could be seen either below or above the text line (Refer Figure 6). This is called vertical drift (Cohen, 2013). In order to get accurate data, we need to re-align the fixations with the text line. We used eye-map tool (Tang, Reilly, & Vorstius, 2012) to map the fixations points to text line filtered from step 2 (Refer Figure 7). In addition to mapping, the tool provides data on several eye-movement variables in the form of word and fixation report including fixation durations, fixation counts, saccades, regressions etc.
4. *Computing Reading Score and reading depth-* We used word and fixation report exported from eye-map to compute reading scores using Buscher et al. 2008 (Buscher et al., 2008) reading detection algorithm and reading depth. The word and fixation reports were first imported to SQL server database using SSIS package. The reading scores for each participant in two groups (linear and segmented) were computed by

implementing reading detection algorithm (Refer Figure 2) using stored procedure.

Reading depth for each participant was computed by taking ratio of number of fixated words to total of words in the reading sections of the module.



Figure 6: Fixations detected above the text line from eye-tracker software



Figure 7: Fixations aligned on the text line using EyeMap

H1a and H1b were tested using independent samples t - test in SPSS to compare mean reading scores and mean reading depth scores between control (Linear module) and the treatment (segmented module) groups. Independent samples t - test was picked because -1) Data for the groups was found normally distributed using Kolmogorov-Smirnov and Shapiro-Wilk test ($p > .05$), 2) the two groups were equal, and 3) the two groups were independent samples.

In RQ2, the engagement score for each respondent were calculated as the mean of codes for eight questions. H2 was tested using independent samples t-test. Independent samples t-test was picked because Shapiro-Wilk test showed that scores for student engagement in both the groups (treatment $n=42$, control $n=38$) satisfied the conditions of normal distribution (treatment $p=.593$, control $p=.187$) and homogeneity of variance ($F=2.554$, $p=.114 > 0.05$).

In RQ3, the scores for each category in the software security survey were calculated based on the correct answers out of - 5 for secure coding awareness (integer overflow (2), input validation (2), buffer overflow (1)), 5 for general software security awareness and 9 for ability to apply secure coding knowledge on code segments (integer overflow (3), input validation (3) and, buffer overflow (3)). In computer literacy survey, the scores for each category were calculated based on the correct answers out of - 4 for phishing awareness. To assess students' ability to apply phishing knowledge, qualitative analyses was performed on students' open-ended answers and were grouped in seven categories (See **Table 8**). Each category weighted a score of 1.

Category	Description	Score
Links	There are suspicious links in an email/ Never click on the link/ Type the website address in the browser	1
Content	There is suspicious content/suspicious words, phrases or sentences	1
Errors	There are grammatical or spelling errors in an email	1
Greetings	Email starts with generic greetings	1
Pop-ups/Attachments	Email contains any pop-up boxes or attachments	1
Urgency	Urgency of an email	1
Personal Information	Email asks for personal information/ never give your personal information	1

Table 8: Score evaluation matrix for ability to phishing knowledge

H3a and H3c were tested using Wilcoxon-Signed-Ranks non-parametric test to

compare the mean rank of scores between pre-survey and post-survey scores. Non-parametric test was picked because – 1) the Kolmogorov-Smirnov and Shapiro-Wilk test showed that the scores are not normally distributed ($p < 0.05$), and 2) the groups (pre and post) were related samples.

H3b, H3d and H3g were tested using Mann-Whitney non-parametric test to compare the mean rank of the scores in two groups (control and treatment). Non-parametric test was picked because – 1) n for the group were not equal, 2) Kolmogorov-Smirnov and Shapiro-Wilk test showed that the scores are not normally distributed ($p < 0.05$), and 3) the two groups were independent samples.

H3e was tested using paired sample t-test to compare the mean of pre-test scores and mean of post-test scores in control and treatment group. Parametric test was picked because 1) Kolmogorov-Smirnov and Shapiro-Wilk test showed that the scores are normally distributed ($p > 0.05$), and 2) the two groups were paired samples.

H3f was tested using independent sample t-test to compare the mean of phishing scores in control and treatment group. Parametric test was picked because 1) Kolmogorov-Smirnov and Shapiro-Wilk test showed that the scores are normally distributed ($p > 0.05$), and 2) the two groups were independent samples.

In RQ4, the usability score for each respondent were calculated as the mean of codes for eighteen questions. H4 was tested using Mann-Whitney U-test. Mann-Whitney non-parametric test was used to compare the mean rank of the scores in two groups (control and treatment). Non-parametric test was picked because – 1) n for the group were not equal, 2) Kolmogorov-Smirnov and Shapiro-Wilk test showed that the scores

are not normally distributed ($p < 0.05$), and 3) the two groups were independent samples.

3.9 Limitations and Assumptions

This research was conducted with the acknowledgment of the following limitations:

- 1 The selection of subjects was limited to 19 (RQ1), 80 (RQ2), 53 (RQ3) and 538 (RQ4) students in the Computer Science courses at Towson University during fall 2014 – spring 2016. The sample was a sample of convenience for RQ2, RQ3 and RQ4 and introduced bias.
- 2 The courses included different teachers each semester and although materials were distributed uniformly, the study was limited due to possible variances in teaching style of the different instructors.
- 3 This report used student questionnaires. Although it is assumed that students answered questions truthfully and honestly, this study was limited due to the individual differences in student self-assessment.
- 4 The questionnaire could only be administered to students who actually attended class when the test was given; there is no data for students who did not attend.
- 5 The eye-tracking device being economically expensive, only one device could be used, resulting in less students participating in the select study due to time constraints.
- 6 One big challenge using eye-tracking was eye-calibration where participants wearing eye-glasses or having eye disorders were difficult to calibrate, resulting in their exclusion from the study.
- 7 Using eye-tracking, it was difficult to restrict participants to be in eye-trackers range

for a long period without much physical movement, resulting in loss of data.

3.10 Institutional Review Board

Approval by Towson University's Institutional Review Board (IRB) for research involving the use of Human Participants was granted for under Exemption Number 09-0xii. The research was exempt from general Human Participants requirement according to 45 CFR 46.101(b)(2). As noted earlier, participation in the study was voluntary, anonymity of the participant was insured, and the participant was fully informed of the research project.

3.11 Summary

This study evaluated the effectiveness of segmented and interactive learning modules in reducing content skipping, increasing student engagement, student learning and usability across CS0 courses over the course of four semesters. Two new instruments, to assess student engagement and module usability, were developed. Reliability analysis of new instruments was conducted to assess the internal consistency. The overall reliability with a value of .74 was found in student engagement and 0.97 in usability instrument. Study approval by Towson University's Institutional Review Board (IRB) for Research Involving the Use of Human Participant was granted under Exception Number 09-0xii.

4. System Implementation

The research included enhancing linear modules to incorporate e-learning design principles of segmentation and interactivity using security injections @Towson cybersecurity modules. The enhanced modules were developed using django Model – View Controller (MVC) framework, hosted on a linux (Centos) server at Towson University. This chapter describes the module design and system implementation.

4.1 Module Design

The modules were designed on two major e-learning principles – segmentation and interactivity. The segmentation is applied by breaking the original linear content into short segments and presenting each segment one at a time. This ensures -1) less page scrolling, 2) less chances of losing the context of text while returning to current page from external pages, 3) less perception of document length, and 4) easier processing of text leading to less content skipping. Interactivity is applied by adding assessment questions that return feedback on submit (dialoguing) and answer-until-correct (controlling), on each segment. This ensures- 1) students read the content, answer assessment questions and receive feedback, 2) students cannot proceed to next segment until answers submit in previous section are correct, and 3) students remain engaged (See **Figure 8**).

Modules contain assessment questions in the form of MCQs to ensure retention and applying of knowledge (deep and surface learning); short answer and constructed response to ensure applying of knowledge (deep learning). The feedback-type on

assessments are knowledge response (KR) and elaborate feedback (EF). Elaborate feedback are provided only after third submit.

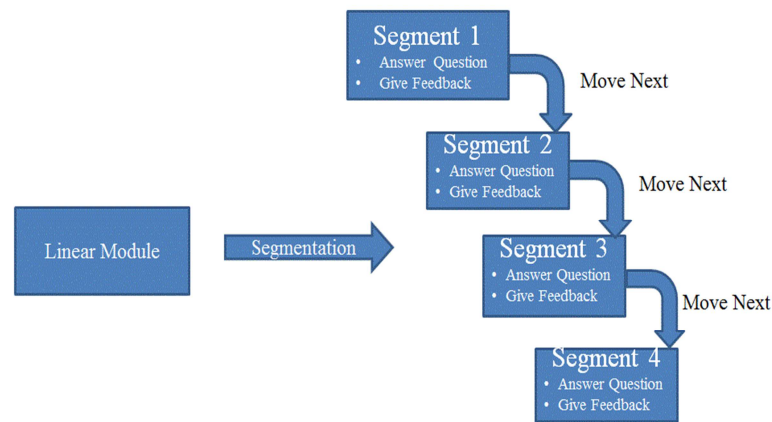


Figure 8: Segmentation & Interactivity Module Design

4.1.1 Security Injections@Towson Modules

The original modules are developed on the cognitive learning principles of Bloom's taxonomy, adopt a uniform structure (See **Figure 9**). Each module begins with a background section to describe the problem with examples, followed by Code Responsibly (includes methods to avoid security issues), a laboratory assignment with a security checklist, and discussion questions sections. The module content is presented as hypertext on a single webpage (B. Taylor & Kaza, 2011a). The module structure is designed to help students to first understand the problem through background and code responsibly sections, then remember it through laboratory assignments and apply the concepts learned, through discussion questions (B. Taylor & Kaza, 2011a). In order to complete the module, students have to read the background section followed by code

responsibly section and then complete the laboratory assignment with security checklist and discussion questions. This ensures the implementation of active learning in security injection modules (B. Taylor & Azadegan, 2007).

How can you learn to prevent this?

VIEW MODULE CONTENT

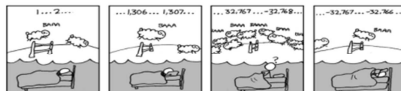
USE THIS MODULE
Give the link to students

Integer Error – “You Can’t Count That High” – CS0

Background

Summary:

Integer values that are too large or too small may fall outside the allowable range for their data type, leading to undefined behavior that can both reduce the robustness of your code and lead to security vulnerabilities.



Source: CNET News Group (http://www.cnet.com/)

Description:

Declaring a variable as type **int** allocates a fixed amount of space in memory. Most languages include several integer types, including **short**, **int**, **long**, etc., to allow for less or more storage. The amount of space allocated limits the range of values that can be stored. For example, a 32-bit **int** variable can hold values from -2147483648 through 2147483647.

Input or mathematical operations such as addition, subtraction, and multiplication may lead to values that are outside of this range. This results in an integer error or overflow, which causes undefined behavior and the resulting value will likely not be what the programmer intended. Integer overflow is a common cause of software errors and vulnerabilities.

Risk – How Can It Happen?

An integer error can lead to unexpected behavior or may be exploited to cause a program crash, corrupt data, lead to incorrect behavior, or allow the execution of malicious software.

Example of Occurrence:

1. There is a Facebook group called “If this group reaches 4,294,967,296 it might cause an integer overflow.” This value is the largest number that can fit in a 32-bit unsigned integer. If the number of members of the group exceeded this number, it might cause an overflow. Whether it will cause an overflow or not depends upon how Facebook is implemented and which language is used – they might use data types that can hold larger numbers. In any case, the chances of an overflow seem remote, as roughly 2/3 of the people on earth would be required to reach the goal of more than 4 billion members.

2. On December 31, 2004, Conair airlines was forced to ground 1,400 flights after its flight crew scheduling software crashed. The software used a 16-bit integer (max 32,768) to store the number of crew changes. That number was exceeded due to bad weather that month which led to numerous crew reassignments.

3. On June 4, 1999 an unmanned defense 3 rocket launched by the European Space Agency exploded just forty seconds after its lift-off from Kourou, French Guiana. Ariane explosion The rocket was on its first voyage, after a decade of development costing 5.7 billion. The destroyed rocket and its cargo were valued at \$500 million. A board of inquiry investigated the cause of the explosion and in two weeks issued a report. It turned out that the cause of the failure was a software error in the inertial reference system. Specifically a 64-bit floating point number relating to the horizontal velocity of the rocket with respect to the platform was converted to a 16-bit signed integer. The number was larger than 32,767, the largest integer storable in a 16-bit signed integer, and thus the conversion failed.

Code Responsibility – How Can I Avoid An Integer Error?

1. **Know your limits:** Familiarize yourself with the ranges available for each data type. With languages such as C and C++, the sizes of the data types are machine and compiler dependent. Run Program 1 below to help you learn the sizes.

2. **Choose your data types wisely:** Many programming languages contain multiple data types for storing integer values. If you have any concerns about the integer values that you will be using, learn about the options available in the language you are using, and choose integer types that are large enough to hold the values you will be using.

3. **Validate your input:** Check input for range and reasonableness before conducting operations. (More on this later.)

Laboratory Assignment

Program 1

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

    cout << "For this compiler: " << endl;
    cout << "largest int: " << INT_MAX << endl;
    cout << "largest integer is: " << INT_MAX << endl;
    cout << "smallest integer is: " << INT_MIN << endl;

    cout << "Enter two integer values: " << endl;
    int x, y;
    cout << "Enter " << x << " and " << y << endl;

    /*comment the next lines for now
    int result = x * y;
    cout << "Your number times 10 is: " << result << endl;
    result = x * 3;
    cout << "The sum of your numbers is: " << result << endl;
    result = x + y;
    cout << "The product of your numbers is: " << result << endl;
    */
    return 0;
}
```

Lab Questions:

- Type the program above and compile. Run and enter reasonable integer values.
- Look at the output. What is the largest possible value type **int** the program can handle?
- Remove the comment lines: `/*` and `*/`. Compile and run again.
- Try inputting a large number to see if you create an error. Did you get an error when you type in a million (1,000,000)? 2 billion (2,000,000,000)? 10 billion (10,000,000,000) for both values? Don't type the commas when you enter the numbers.
- Complete the security checklist for this program. (Print the checklist).
- What happens when the result of an operation exceeds the range of the **int** type? Explain.
- How could addition result in an integer overflow?
- How could multiplication result in an integer overflow?
- What operation is most likely to cause an integer overflow?

*Copying and pasting programs may result in syntax errors and other inconsistencies. It is recommended you type each program.

Security Checklist

Security Checklist	
Vulnerability: Integer Errors Course: CS0	
Check each line of code	Completed
1. Underline each occurrence of an integer variable.	
For each undefined variable:	
2. Mark with a V any input operations that assign values to the variable.	
3. Mark with a V any mathematical operations involving the variable.	
4. Mark with a V any assignments made to the variable.	
Highlighted areas indicate vulnerabilities!	

Discussion Questions

- What is the largest possible value of type **int**? Explain your answer using the information you read in the Background section.
- What happens when the result of an operation on values of type **int** exceeds this value? Explain.

Further Work (optional – check with your instructor if you need to answer the following questions)

- Look up the following info:
 - What is the population of the US?
 - What is the population of the world?
 - What is the national debt?
- For which of the above would the **int** data type be a problem?
- Discuss the Conair problem described above. What are the repercussions of such a problem?

Go To Top



Copyright © Towson University

Figure 9: Original Security Injections Linear Module Format

In enhanced modules, the original linear module content is broken into four segments that begin with a background segment to describe the problem with examples, followed by a “Code Responsibly” segment (that includes methods to avoid security issues), a laboratory assignment segment with a security checklist, and discussion questions segment.

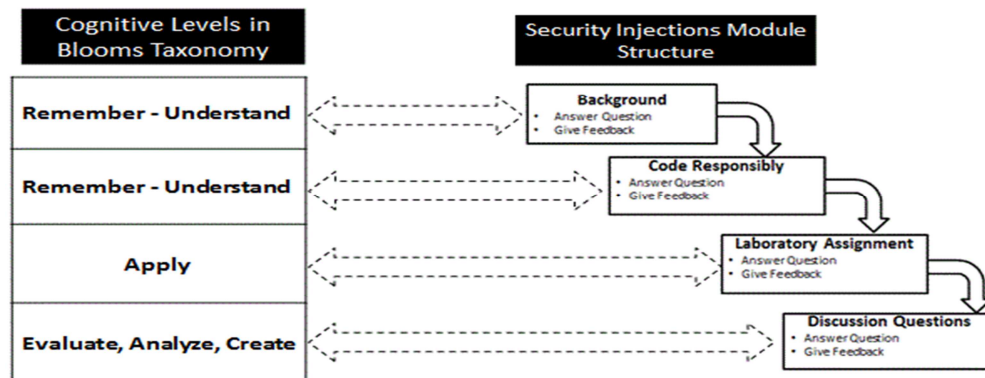


Figure 10: Mapping cognitive levels in Blooms Taxonomy to enhanced learning modules

The module structure is designed to help students to first remember the and understand the problem through the background and code responsibly segments; apply it through the laboratory assignments including security checklist; and evaluate, analyze and create the concepts through discussion questions (See **Figure 10**).

In the background and code responsibly segments, students are required to go through the content and answer a set of checkpoint questions. Each question provides immediate feedback on submit. The student cannot advance to the next section until all questions are answered correctly. In the laboratory assignment and discussion question,

students answer text-based, multiple choice questions, and identify vulnerabilities based on a security checklist.

4.2 System Development

To implement a system, several solutions were considered (including writing the system from scratch) before determining that a modified version of Stanford University's class2go web-based application (<https://github.com/Stanford-Online/class2go/>) was most appropriate. Class2go is built using the Django framework. Django is a high level Python based Model-View-Controller (MVC) framework. In Django terminology, model, which is usually in a models.py file, defines data in Python and syncs with the databases, which typically contains a relational database like Mysql, Sqlite, PostgreSQL etc., template is similar to view and returns html page, and view is similar to controller which performs the requested action and modifies the data (See **Figure 11**)

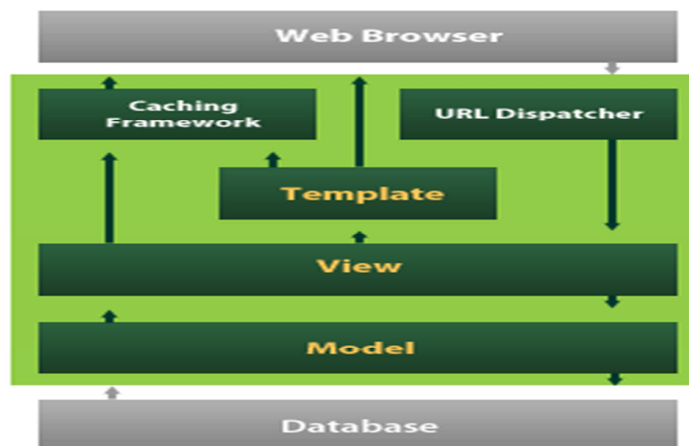


Figure 11: Django MVC architecture

When URL in a web browser is entered, the URL dispatcher, which is a `urls.py` file, maps it to view function in `views.py` file and evokes it. The view function performs actions like reading or writing to the database. The view, after performing the requested task, returns an HTTP object through a template (usually `.html` file) to the web browser. Class2go is a well-tested open-source framework that provides core functionality, course creation, test administration, and some components for auto-grading. The application creates modules and segments within those modules. Each segment in a module is auto-graded using built-in functionality for text and multiple choice questions. Class2go uses MySQL database to store data. The data includes user information, HTML/XML of modules and scores earned by students in each module.

In class2go environment, the content and formative assessment including multiple choice questions and constructed response (short answer and essay) question sets was created using HTML. The student submitted response to multiple choice questions and constructed response questions, and provide knowledge response (KR) and elaborate feedback (EF) was done using XML. To verify answers for constructed response questions, regular expressions, were written, which match keywords, stored in xml, with the answers, submitted by students. The set of keywords are generated based on the module content. Elaborate feedback is presented to students only after 3rd attempt. See **Figure 12** for MCQ HTML code snippet, **Figure 13** for MCQ XML code snippet, Fig 5.5 for constructed response HTML code snippet, **Figure 14** for constructed response XML code snippet and Fig 5.7 for elaborative feedback.

```

<h3>Answer the following questions:</h3>
<div id="problem_1" title="CS1 Java Integer Error Code Responsibly
Section Question 1"
data-report="CS1-Java-Integer-Error-Code-Responsibly-Section-Question-
1" class="question">
  <h3 class="questionNumber"><span>Question 1: </span></h3>
  <h4>How can you avoid an integer error in your program?</h4>
  <fieldset name="Q1_MC1" data-report="Sources-of-input-for-
programs">
    <label for="Q1_MC1_1">
      <input value="1" data-report="CS1-Java-Integer-Error-
Code-Responsibly-Section-Question-1_option1" id="Q1_MC1_1"
name="Q1_MC1" type="checkbox">Know the smallest and largest allowable
values for each data type in the programming language you are us-
ing</label>
    <label for="Q1_MC1_2">
      <input value="2" data-report="CS1-Java-Integer-Error-
Code-Responsibly-Section-Question-1_option2" id="Q1_MC1_2"
name="Q1_MC1" type="checkbox">Always pick float or double as the data
type for numbers</label>
    <label for="Q1_MC1_3">
      <input value="3" data-report="CS1-Java-Integer-Error-
Code-Responsibly-Section-Question-1_option3" id="Q1_MC1_3"
name="Q1_MC1" type="checkbox">Check your input for reasonable values
before conducting mathematical operations</label>
  </fieldset>
  (Hint: read the code responsibly section above to answer this
question.)
</div>

```

Figure 12: MCQ Html code snippet

```

<exam_metadata>
  <question_metadata id="problem_1" data-report="CS1-Java-Integer-
Error-Background-Section-Question-1">
    <response data-report="Sources-of-input-for-programs"
name="Q1_MC1" answertype="multiplechoiceresponse">
      <choicegroup type="MultipleChoice">
        <choice correct="true" data-report="CS1-Java-Integer-
Error-Code-Responsibly-Section-Question-1_option1" value="1">
        </choice>
        <choice correct="false" data-report="CS1-Java-Integer-
Error-Code-Responsibly-Section-Question-1_option2" value="2">
        </choice>
        <choice correct="true" data-report="CS1-Java-Integer-
Error-Code-Responsibly-Section-Question-1_option3" value="3">
        </choice>
      </choicegroup>
    </response>
  </question_metadata>
</exam_metadata>

```

Figure 13: MCQ xml snippet

```

<div data-report="cs1_java_buffer_overflow_discussion_questions_problem_1"
id="problem_1"
class="question">
  <h3 class="questionNumber">Question 1</h3>
  <p>Describe the buffer overflow problem. <br/>
    <textresponse
      data-report="Discussion Question problem 1">
    <textarea
id="cs1_java_buffer_overflow_discussion_questions_problem_1"
name="cs1_java_buffer_overflow_discussion_questions_problem_1"
style="height:100px;width:80%;"></textarea>
    </textresponse>
  </p>
</div>

```

Figure 14: Constructed Response HTML Code snippet

```

<question_metadata data-report="cs0_cpp_buffer_overflow_discussion_questions_problem_1"
id="problem_1">
  <solution>
    <div class="detailed-solution">
      <p>Answer Set:</p>
      <p>when a program attempts to access a
value that is outside of the specified data buffer will cause buffer
overflow</p>
    </div>
  </solution>
  <response
    answer="((?=.*?\b(?:b[ufer]*)\b)(?=.*?\b(?:o[ver]*\s*f[low]*)\b)
    (?=.*?\b(?:d[ata]*)\b)(?=.*?\b(?:out[side]*)\b))|(buf(?:ufer)*)|(o[ver]
    *\s*f[low]*)|(o[ut]*\s*s[ide]*)|(num(ber)?|v[al]*(ue)?)" data-report="Discussion Question problem 1" answertype="regexresponse"
    name="cs0_cpp_buffer_overflow_discussion_questions_problem_1_1"
    id="cs0_cpp_buffer_overflow_discussion_questions_problem_1_1">
    <responseparam flag="IGNORECASE" />
    <responseparam flag="MULTILINE" />
  </response>
</question_metadata>

```

Figure 15: Constructed Response XML code snippet a) Bold: shows elaborated feedback b) Gray: shows regular expression

The class2go environment is installed on a 64-bit linux (centos 6.5) virtual server hosted by Towson University. The system implementation, including server set up,

class2go installation and module development, began in summer 2013. A total of 8 developers, undergraduate and graduate students, have contributed to the system development over seven semesters. Considering the vastness of the system, a version control system was set up using subversion. In addition, a development server was installed in order to test the code revisions before deploying the code to production server (See **Figure 16**).

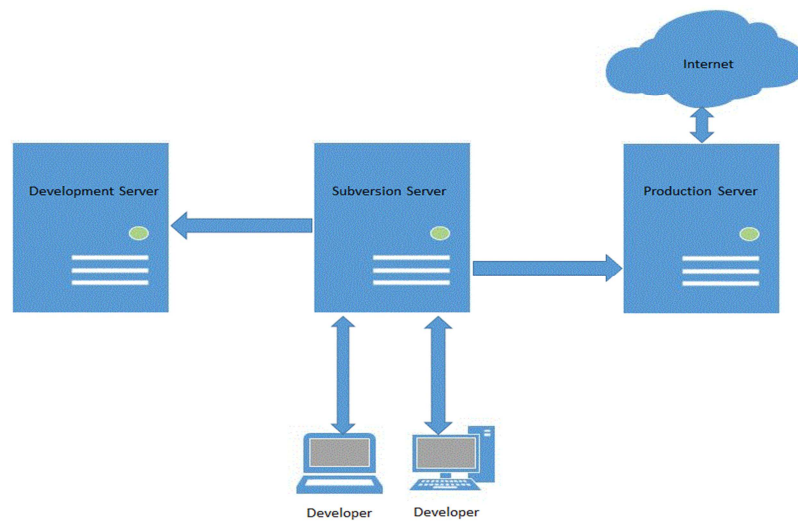


Figure 16: Systems development process

Approximately forty modules have been developed so far (refer **Figure 17**).

Figure 18– Figure 23 present screen shots of integer error module in CS0 course in C++.

	CS0	CS1	CS2
Integer Error	C++ Java Pseudocode	C++ Java	C++ Java
Input Validation	C++ Java Python Pseudocode	C++ Java Python	C++ Java
Buffer Overflow	C++ Java Python Pseudocode	C++ Java Python	C++ Java
Software Development Life Cycle	C++ Java Python		
Best Practices for Secure Variables		Java	
Encapsulation			C++ Java
Exception Handling			C++ Java

Computer Literacy	Passwords	Module
	Phishing	Module
	Cryptography	Module
	Social Networking Security	Module

Figure 17: List of security injection modules

Security Injections, CPP CS0 - Integer Error

1. Background

2. Code Responsibility

3. Laboratory Assignment

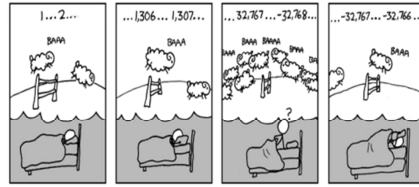
4. Discussion Questions

Integer Error - "You Can't Count That High" - CS0

Background

Summary:

Integer values that are too large or too small may fall outside the allowable range for their data type, leading to undefined behavior that can both reduce the robustness of your code and lead to security vulnerabilities.



Description:

Declaring a variable as type `int` allocates a fixed amount of space in memory. Most languages include several integer types, including `short`, `int`, `long`, etc., to allow for less or more storage. The amount of space allocated limits the range of values that can be stored. For example, a 32-bit `int` variable can hold values from -2^{31} through $2^{31}-1$.

Input or mathematical operations such as addition, subtraction, and multiplication may lead to values that are outside of this range. This results in an integer error or overflow, which causes undefined behavior and the resulting value will likely not be what the programmer intended. Integer overflow is a common cause of software errors and vulnerabilities.

Risk - How Can It Happen?

An integer error can lead to unexpected behavior or may be exploited to cause a program crash, corrupt data, lead to incorrect behavior, or allow the execution of malicious software.

Example of Occurrence:

1. There is a Facebook group called "If this group reaches 4,294,967,296 it might cause an integer overflow." This value is the largest number that can fit in a 32-bit unsigned integer. If the number of members of the group exceeded this number, it might cause an overflow. Whether it will cause an overflow or not depends upon how Facebook is implemented and which language is used - they might use data types that can hold larger numbers. In any case, the chances of an overflow seem remote, as roughly 2/3 of the people on earth would be required to reach the goal of more than 4 billion members.
2. On December 25, 2004, Comair airlines was forced to ground 1,100 flights after its flight crew scheduling software crashed. The software used a 16-bit integer (max 32,768) to store the number of crew changes. That number was exceeded due to bad weather that month which led to numerous crew reassignments.



Answer the following questions:

Question 1:

Declaring a variable as type `integer`:

- ☒ X Allocates an infinite amount of storage
- ☐ Allocates a fixed amount storage

(Hint: read summary and description sections to answer this question.)

Question 2:

An integer error in C++ or Java causes:

- ☐ a syntax error
- ☐ the program to correct itself
- ☒ unexpected behavior

(Hint: read the risk section above to answer this question.)

Incorrect! Try Again



This project is supported by the National Science Foundation under grants DUE-1241738 and DUE-0817287. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Powered by a modified version of Cien20x

Figure 18: Enhanced security injection module (Background section)

Security Injections, CPP CS0 - Integer Error

1. Background

2. Code Responsibly

3. Laboratory Assignment

4. Discussion Questions

Code Responsibly - How can I avoid an Integer Error?

1. *Know your limits:* Familiarize yourself with the ranges available for each data type. With languages such as C and C++, the sizes of the data types are machine and compiler dependent. Run Program 1 below to help you learn the sizes.

2. *Choose your data types wisely:* Many programming languages contain multiple data types for storing integer values. If you have any concerns about the integer values that you will be using, learn about the options available in the language you are using, and choose integer types that are large enough to hold the values you will be using.

3. *Validate your input:* Check input for range and reasonableness before conducting operation.(More on this later)

Answer the following questions:**Question 1:****How can you avoid an integer error in your program?****There are multiple correct answers, try again.**

- ☒ Know the smallest and largest allowable values for each data type in the programming language you are using
- ☒ Always pick float or double as the data type for numbers
- ☐ Check your input for reasonable values before conducting mathematical operations

(Hint: read the code responsibly section above to answer this question.)

Incorrect! Try Again



This project is supported by the National Science Foundation under grants DUE-1241730 and DUE-1061720. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Powered by a modified version of Class2Go

Figure 19: Security injection module (code responsibly section)

Security Injections, CPP CS0 - Integer Error

1. Background 2. Code Responsibility 3. Laboratory Assignment 4. Discussion Questions

Laboratory Assignment

(Note: refer Background and Code Responsibility sections to answer the Laboratory Assignment.)

STEP 1: Type program 1 and compile. Run and enter reasonable integer values.

```

Program 1
#include <iostream>
#include <limits>
using namespace std;
int main ()
{
    int i;
    int j;

    cout << "For this compiler: " << endl;
    cout << "Integer size: " << 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;

    //compute the next line for now
    int result = i * j;
    cout << "Your number times 10 is: " << result << endl;
    result = i * j;
    cout << "The sum of your numbers is: " << result << endl;
    result = i * j;
    cout << "The product of your numbers is: " << result << endl;

    return 0;
}

```

Question 1

Look at the output. What is the largest possible value of type `int` the program can handle?
STEP 2: Remove the comment lines: `//` and `/*`. Compile and run again.

Question 2

Try entering a large number to see if it leads to an error. Did you get an error when you type in the following numbers for both input values? Don't type the commas when you enter the numbers.

a) 1 million (1,000,000)?

- ☐ Yes
☐ No

b) 2 billion (2,000,000,000)?

- ☐ Yes
☐ No

c) 10 billion (10,000,000,000)?

- ☐ Yes
☐ No

STEP 3: Complete the Security Checklist

[Click to see how a checklist works](#)

Question 3

```

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

    cout << "For this compiler: " << endl;
    cout << "Integer size: " << 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 * j;
    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 numbers is: " << result << endl;

    return 0;
}

```

Vulnerability: Integer-Overflow: CS0001	Completed?
Check each line of code	<input checked="" type="checkbox"/>
1. Click each declaration of an integer variable.	<input checked="" type="checkbox"/>
For each variable from 1:	
2. Click all input operations that assign values to the variable.	<input checked="" type="checkbox"/>
3. Click all mathematical operations involving the variable.	<input checked="" type="checkbox"/>
4. Click all assignments made to the variable.	<input checked="" type="checkbox"/>
Highlighted areas indicate vulnerabilities	<input checked="" type="checkbox"/>

Question 3:

Which of the following operations can lead to an integer error?

- ☐ Addition
☐ Subtraction
☐ Multiplication

Question 4

Which of the operations listed in question 3 is most likely to cause an integer error?

Submit Answers



This project is supported by the National Science Foundation under grants DUE-1241738 and DUE-1661727. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Powered by a modified version of Causalis

Figure 20 : Security injection module (Laboratory assignment section)

Security Injections, CPP CS0 - Integer Error

1. Background 2. Code Responsibility 3. Laboratory Assignment 4. Discussion Questions

Discussion Questions

(Note: refer Background and Code Responsibility sections to answer the Discussion Questions.)

Question 1

What is the largest possible value of type int? Explain your answer using the information you read in the Background section.

Question 2

What happens when the result of an operation on values of type int exceeds this value? Explain.

Further Work (optional - check with your instructor if you need to answer the following questions)

Question 1

Look up the following info:

What is the population of the US?

What is the population of the world?

What is the national debt?

Question 2:

For which of the above would int data type be a problem:

- ☐ Population of the US
- ☐ Population of the world
- ☐ National Debt

Question 3

Discuss the Comair problem described in background section. What are the repercussions of such a problem?

Submit Answers



This project is supported by the National Science Foundation under grants DUE-1241738 and DUE-0817287. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Powered by a modified version of Cms2Go

Figure 21: Enhanced Security injection modules(Discussion question section)

```

#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 22: Enhanced security injection module (auto-graded security checklist)

Which of the operations listed in question 3 is most likely to cause an integer error?

multiplication

Hide Explanation

Answer Set

Multiplication can increase the value of the integer variables dramatically and results in larger numbers that are more likely to overflow.

Figure 23: Enhanced security injections module (Elaborative explanation)

4.3 Summary

Approximately forty enhanced (segmented-interactive) security injection learning modules were developed over five semesters, using django framework from class2go.

The modules were hosted on a linux (centos) server at Towson University. In addition, considering the vastness and multiple developers working on the project, a development server and version control system was installed to keep track of the code revisions.

5. Results

The research study examined the effectiveness of the enhanced (segmented – interactive) modules over linear modules on students’ content skipping, student engagement, student learning and module usability. This chapter first puts forth the research questions and hypotheses, and discusses the results of four studies.

5.1 RQ1

RQ1: Can the use of learning modules with segmentation reduce content skipping as compared to linear modules?

RQ1a: Can the use of learning modules with segmentation show significantly higher reading scores as compared to linear modules?

H1a: The mean reading scores in treatment group will be significantly higher than the mean score for control group.

The reading scores for each participant was calculated using (Buscher et al., 2008) reading detection algorithm (Refer **Figure 2**). The mean reading scores for the treatment group (1981.9) and mean reading scores for the control group (1186.4) were found to be statistically significant at the 95% level ($p < .05$, $p = 0.03$). This implies that students read significantly more using segmented modules as compared to linear modules. This leads us to accept H1 (Refer Figure 24).

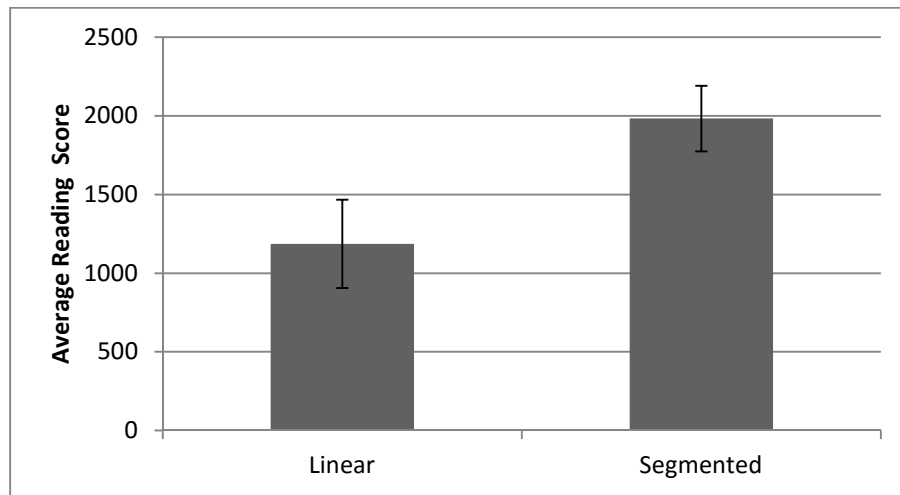


Figure 24: Average reading scores in control and treatment groups

RQ1b: Can the use of learning modules with segmentation show significantly higher reading depth as compared to linear modules?

H1b: The mean reading depth in treatment group will be significantly higher than the mean reading depth in control group.

The reading depth for each participant was calculated based on (number of words fixated / total number of words) The mean reading depth scores for the treatment group (0.56) and mean reading depth scores for the control group (0.37) were found to be statistically significant at the 95% level ($p < .05$, $p = 0.04$). The mean reading depth scores were higher using segmented modules compared to linear modules. This implies that students using segmented modules covered more text as compared to linear modules. This leads us to accept H2 (Refer **Figure 25**).

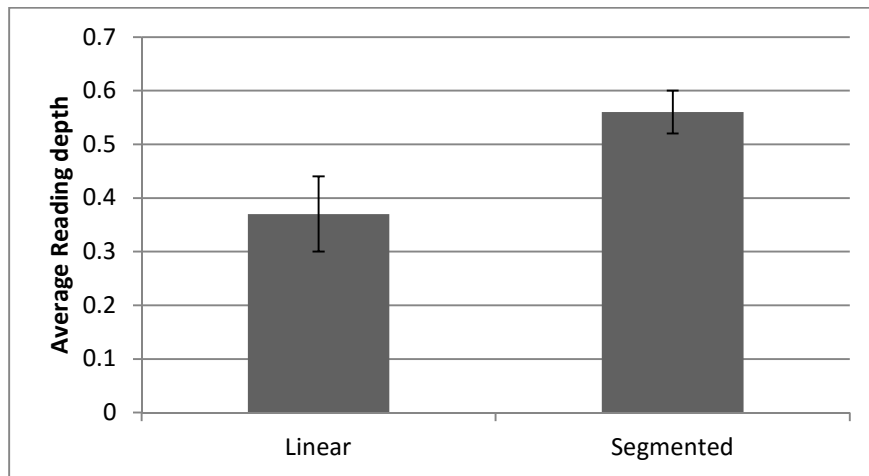


Figure 25 : Average reading depth in control and treatment groups

5.2 RQ2

RQ2: Can the use of learning modules with segmentation and interactivity increase student engagement as compared to linear modules?

H2: The mean of survey scores for student engagement in the treatment group will be significantly higher than the mean of the survey scores for student engagement in the control group.

In the survey results, the mean score for the treatment group ($n=42$, $\text{mean}=3.43$) was found to be significantly higher at 95% level ($t=-2.265$, $p=0.026$) than the mean score for the control group ($n=38$, $\text{mean}=3.19$). This implies that students found enhanced (segmented and interactive) modules more engaging than the linear modules (see **Figure 26a**). This leads us to accept H1 and supports research question RQ2. In addition, higher engagement persisted across gender (see **Figure 26b**) and race (see **Figure 26c**).

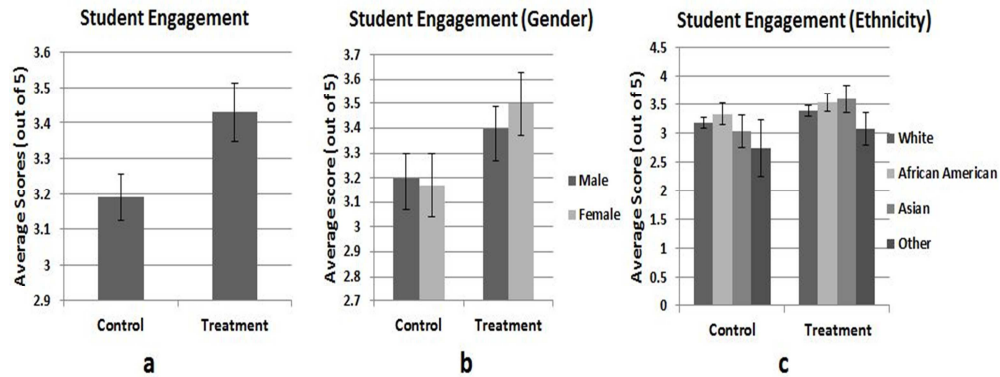


Figure 26: (a) Average student engagement score in treatment and control group (b) Average student engagement score between males and females in treatment and control group (c) Average student engagement score between ethnic groups in treatment and control group

The mean student engagement mean score for the treatment group was found to be higher than the control group (see **Table 9**). In particular, the scores for Q6 and Q8 were found to be statistically significant at 95% level (refer to **Table 9** for survey questions).

	Mean Score	
	Control (n = 38)	Treatment (n = 42)
Q1	3.39	3.48
Q2	2.63	2.74
Q3	2.89	3.07
Q4	3.37	3.45
Q5	3.34	3.33
Q6	3.08	3.79*
Q7	3.50	3.79
Q8	3.29	3.79*
Student Engagement Mean Score	3.19	3.43*
*p < 0.05 (statistically significant at 95% level)		

Table 9: Results of individual engagement survey questions

5.3 RQ3

RQ3: Can the use of modules with segmentation and interactivity increase student learning as compared to linear modules?

RQ3a: Can the use of learning modules with segmentation and interactivity show same learning as compared to linear modules for questions that assess retention of knowledge?

H3a: The post-survey scores for secure coding awareness (measuring retention) will be significantly higher than the pre-survey scores for secure coding awareness in both control and treatment groups.

In the treatment group, the average score for pre-survey (1.5) and the average score for post-survey (4.0) were found to be statistically significant at the 95% level ($p < 0.05$, $z = -4.02$). In the control group, the average score for pre-survey (1.76) and the average score for post-survey (3.81) were found to be statistically significant at the 95% level ($p < 0.05$, $z = -3.78$). This implies that the use of both linear and enhanced (segmented-interactive) modules significantly increased the secure coding awareness (measuring retention) among the students in the post-survey compared to the pre-survey.

In addition, this verifies that any kind of educational intervention has a positive effect on student achievement, and, interactive and non- interactive systems with same content will have the same learning on recall or retention assessments. This leads us to accept H3a (see **Figure 27**).

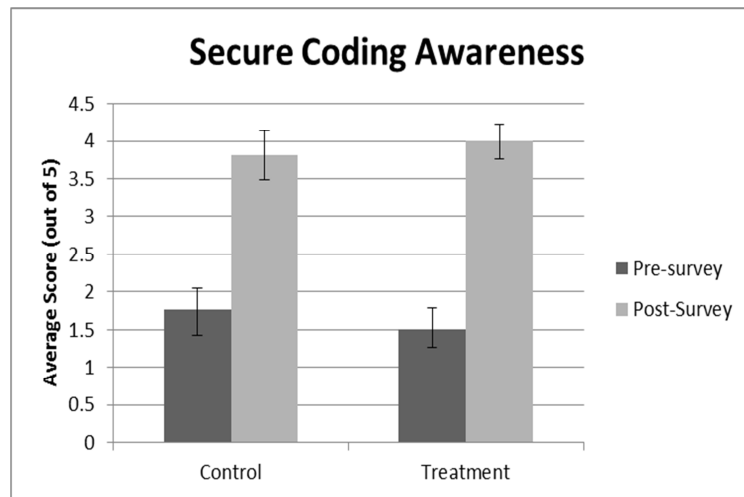


Figure 27: Pre-survey and post-survey scores in the control and treatment groups for security-coding awareness

H3b: The post-survey scores for secure coding awareness (measuring retention) for the treatment and the control group will not be significantly different.

In the post-survey, no significant differences were found between the average scores for the treatment group (4.0) and the control group (3.81). This verifies that interactive and non-interactive systems with same content have same learning on recall or retention assessments. This leads us to accept H3b (see **Figure 27**).

H3c: The post-survey scores for general software security awareness (measuring retention) will be significantly higher than the pre-survey scores for general software security awareness in both control and treatment group.

In the treatment group, the average score for the post-survey (4.21) was significantly higher at the 95% level ($p < 0.05$, $z = -3.056$) than the average score for the pre-survey (3.25). In the control group, the average score for the post-survey (4.14) was also significantly higher at the 95% level ($p < 0.05$, $z = -2.20$) than the average score for

the pre-survey (3.29). This implies that the use of both linear and enhanced (segmented - interactive) modules significantly increased the general software security awareness among the students in the post-survey compared to the pre-survey.

In addition, this verifies that any kind of educational intervention has a positive effect on student achievement, and, interactive and non- interactive systems with same content will have the same learning on recall or retention assessments. This leads us to accept H3c (see **Figure 28**).

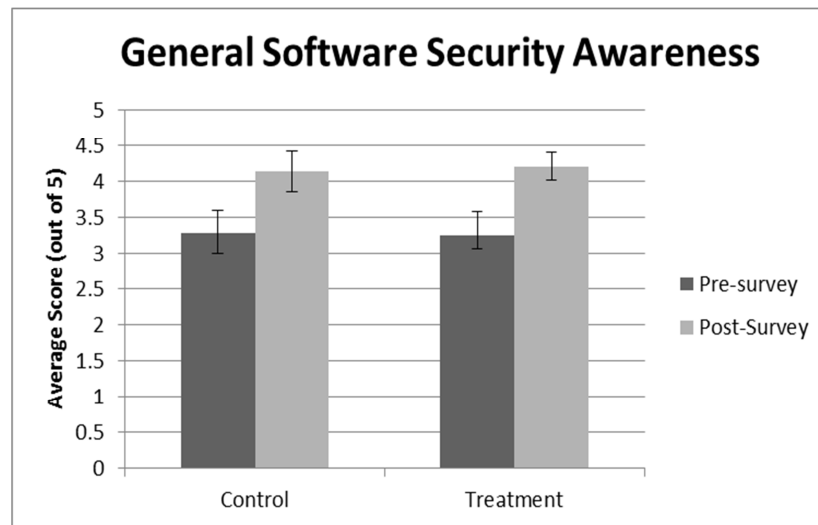


Figure 28: Pre-survey and post-survey scores in the control and treatment groups for general software security awareness

H3d: The post-survey scores for general software security awareness (measuring retention) for the treatment and the control group will not be significantly different.

In the post-survey, no statistically significant differences were found between the average scores for the treatment group (4.21) and the control group (4.14). This verifies that interactive and non-interactive systems with same content have same learning on recall or retention assessments. This leads us to accept H3d. (see **Figure 28**)

H3e: The post-survey scores for phishing awareness (measuring retention) will be significantly higher than the pre-survey scores for phishing awareness in both control and treatment groups.

In the treatment group, the average phishing awareness scores in the post-survey (3.77) was significantly higher at the 95% level ($p < 0.05$, $t = -4.92$) than the average phishing awareness scores in the pre-survey (3.15). In the control group, the average phishing awareness scores in the post-survey (3.72) was also significantly higher at the 95% level ($p < 0.05$, -3.96) than the average score for the pre-survey (3.26). This implies that the use of both linear modules and enhanced modules significantly increased phishing awareness among the students in the post-survey compared to the pre-survey.

In addition, this verifies that any kind of educational intervention has a positive effect on student achievement, and, interactive and non- interactive systems with same content will have the same learning on recall or retention assessments. This leads us to accept H3e (see **Figure 29**)

H3f: The post-survey scores for phishing awareness (measuring retention) for the treatment and the control group will not be significantly different.

In the post-survey, no statistically significant differences were found between the average scores for the treatment group (3.77) and the control group (3.72). This verifies that interactive and non-interactive systems with same content have same learning on recall or retention assessments. This leads us to accept H3f. (see **Figure 29**)

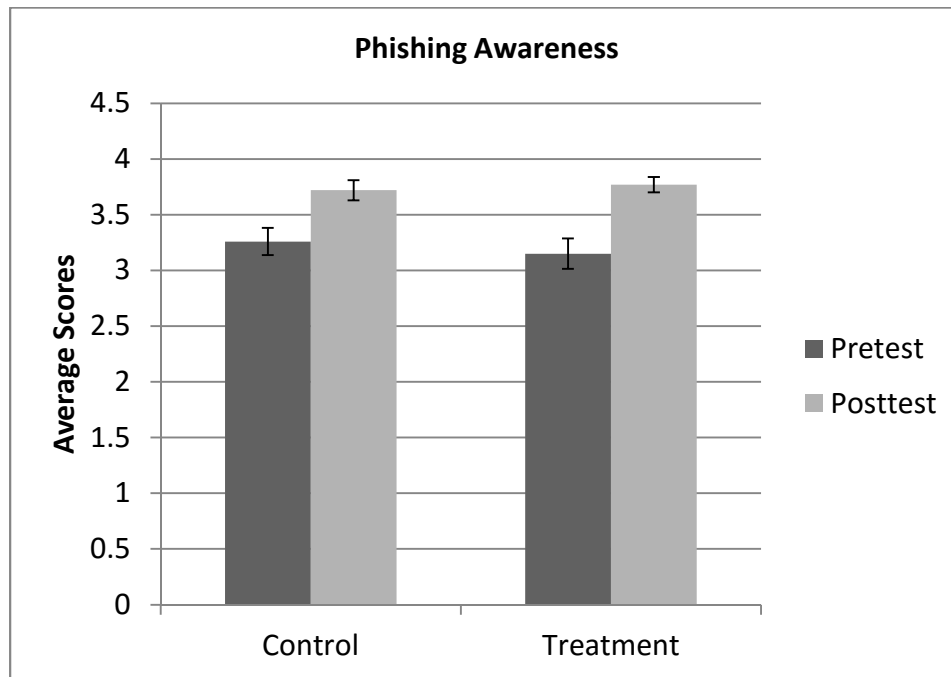


Figure 29: Pre-survey and post-survey scores in the control and treatment groups for general phishing awareness

RQ3b: Can the use of learning modules with segmentation and interactivity show significantly higher learning as compared to linear modules for questions that assess applying of knowledge?

H3g: The scores for ability to apply secure coding knowledge in the treatment group will be significantly higher than the control group.

In the post-survey, the average score for the treatment group (5.59) was found significantly higher at 90% level ($p = 0.07 < 0.10$, $z = -1.80$) than the average score for the control group (4.27). The students who use enhanced modules performed significantly better in identifying security vulnerabilities in three separate code segments than the students who use linear module. This verifies interactive systems show

significantly higher learning on problem-solving (ability to apply) assessments than the non-interactive systems with same content. This leads us to accept H3g (see **Figure 30**).

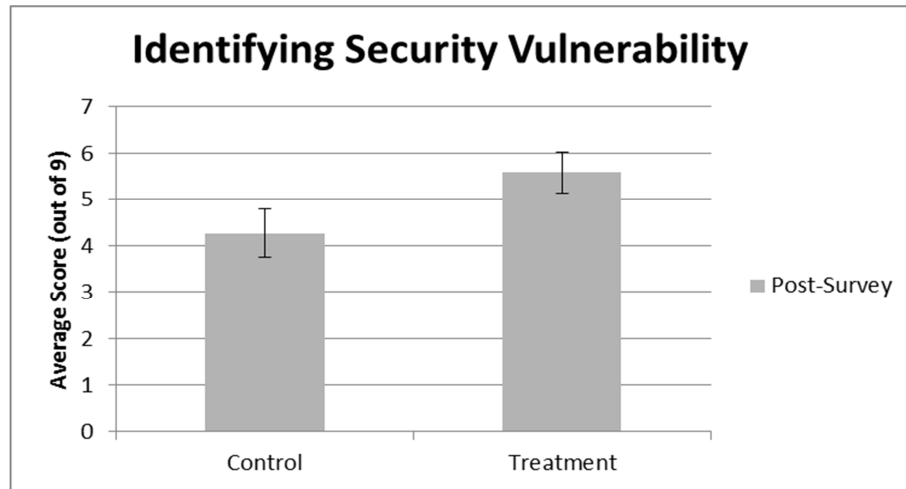


Figure 30: Post-survey scores in the control and treatment groups for ability to identify security vulnerability in three code segments

H3h: The scores for ability to apply phishing knowledge in the treatment group will be significantly higher than the control group.

In the post-survey, the average phishing score for the treatment group (1.67) was found significantly higher at 90% level ($p = 0.08$, $t = -1.73$) than the average score for the control group (1.28). The students who use enhanced modules performed significantly better in identifying a phished email than the students who use linear module. This verifies interactive systems show significantly higher learning on problem-solving (ability to apply) assessments than the non-interactive systems with same content. This leads us to accept H3h (see **Figure 31**).

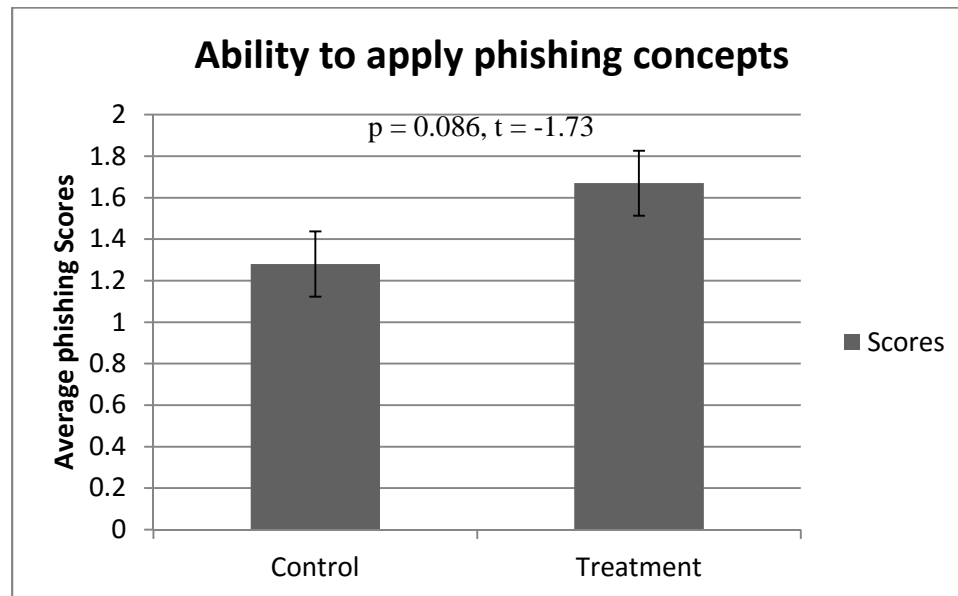


Figure 31: Post-survey scores in the control and treatment groups for ability to identify phishing in an email

5.4 RQ4

RQ4: Are learning modules with segmentation and interactivity significantly more usable than linear modules?

H4: The mean of survey scores for overall usability in the treatment group will be significantly higher than the mean of the survey scores for overall usability in the control group.

In the survey results, the mean score for the treatment group ($n=332$, $\text{mean}=4.17$) was found to be significantly higher at 95% level ($t=-2.265$, $p=0.026$) than the mean score for the control group ($n=206$, $\text{mean}=3.88$). This implies that students found enhanced (segmented-interactive) modules more than linear modules. This leads us to accept H4 and supports research question RQ4 (Refer **Figure 32**).

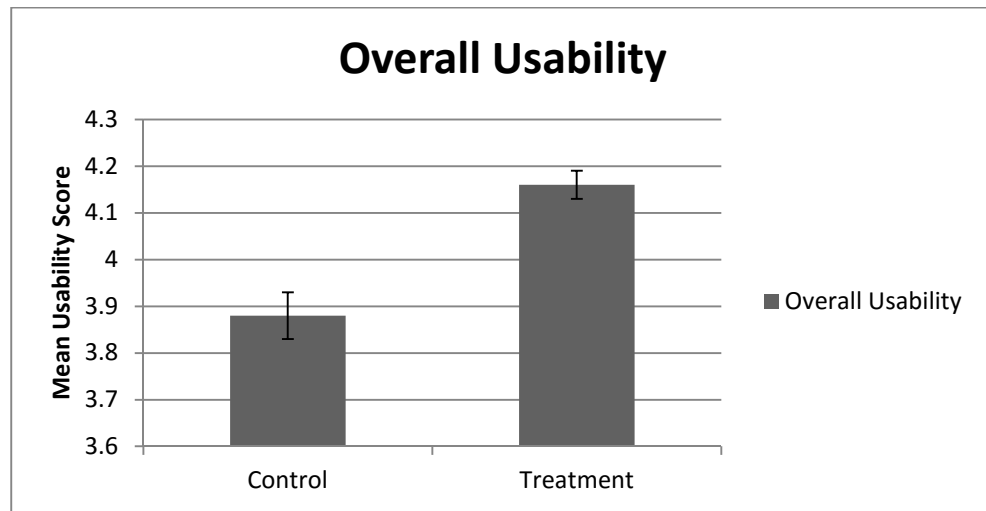


Figure 32: Mean scores for overall usability in control and treatment group

The mean scores for each question in the survey were compared between control and treatment group (see **Table 10**). The results indicate significantly higher scores at 95% level ($p < 0.05$) for each question in segmented-interactive modules.

	Control (n = 206)	Treatment (n = 332)
Q1	3.88	4.13*
Q2	3.90	4.20*
Q3	3.99	4.28*
Q4	4.15	4.32*
Q5	3.95	4.33*
Q6	3.98	4.16*
Q7	4.16	4.39*
Q8	4.18	4.37*
Q9	4.00	4.20*
Q10	3.90	4.21*
Q11	3.70	4.08*
Q12	3.75	4.10*
Q13	3.66	4.04*
Q14	3.72	4.02*
Q15	3.76	4.06*
Q16	3.74	3.98*
Q17	3.84	4.12*
Q18	3.68	4.01*
Usability Mean Score	3.88	4.16*
*p < 0.05 (statistically significant at 95% level)		

Table 10: Mean usability scores for individual questions in a survey

Here the results for integer overflow, input validation and buffer overflow modules between treatment and control group in CS0 and CS1 are compared.

Integer Overflow

In CS0, the mean usability score for the treatment group (n = 72, mean=4.21) was found to be significantly higher at 90% level ($p < 0.10$) than the mean score for control group (n = 67, mean = 3.97). This implies that students in CS0 course found integer overflow modules in enhanced (2.0) version more usable than traditional (1.0) linear version (Refer Fig. 6.). In CS1, the mean usability score for the treatment group (n = 45, mean = 4.12) was found to be significantly higher at 95% level ($p < 0.05$) than the mean

score for control group (n = 33, mean = 3.61). This implies that students found integer overflow modules in enhanced version (2.0) more usable than traditional (1.0) linear version (Refer **Figure 33**)

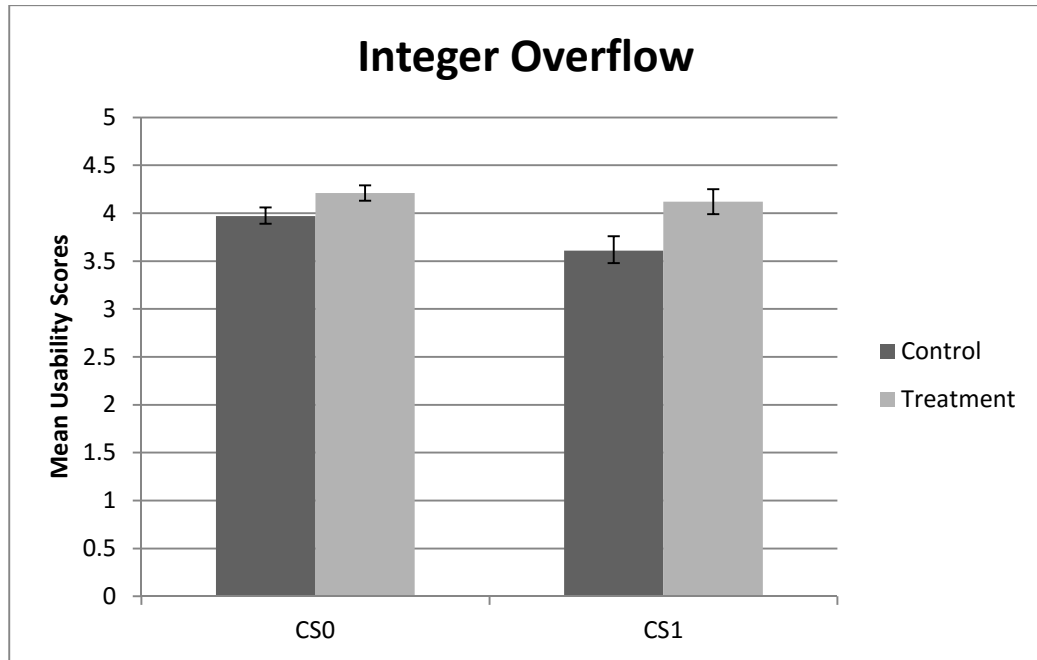


Figure 33 : Mean usability scores in treatment and control group in CS0 and CS1 (integer Overflow)

Input Validation

In CS0, while mean usability score for the treatment group (n = 52, mean=4.16) and control group (n=27, mean = 4.05) were not found to be statistically significant, the mean usability score for the treatment group was found higher. In particular, Q14 was found significantly higher in the treatment group. In CS1, the mean usability score for the treatment group (n = 26, mean = 4.16) was found to be significantly higher at 90% level ($p < 0.05$) than the mean score for control group (n = 14, mean = 3.73). This implies students found input validation module in enhanced version more usable than the

traditional (1.0) linear module in CS1 course. (Refer **Figure 34**)

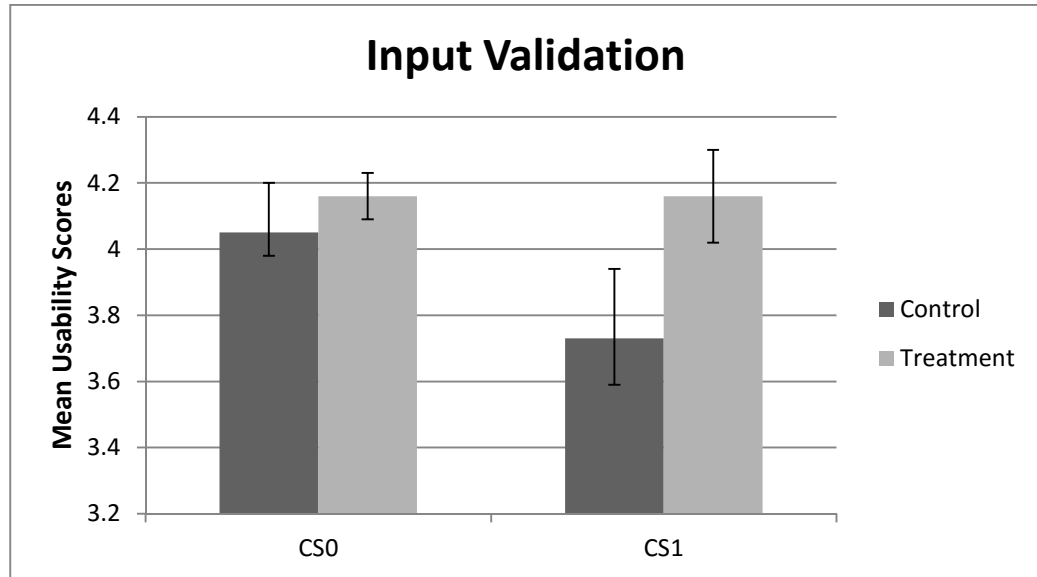


Figure 34: Mean usability scores in treatment and control group in CS0 and CS1 (input validation)

Buffer Overflow

In CS0, while mean usability score for the treatment group ($n = 44$, mean=4.15) and control group ($n=26$, mean = 3.89) were not statistically significant, the mean usability score for the treatment group was found higher. In particular, Q2, Q5, Q6, Q7, Q8 and Q13 were found to be statistically significant. In CS1, the mean usability score for the treatment group ($n = 19$, mean = 4.30) and control group ($n = 9$, mean = 4.40) were not found to be statistically significant. One of the reasons could be small sample size in control group. We plan to further validate the results with large sample size. (Refer **Figure 35**)

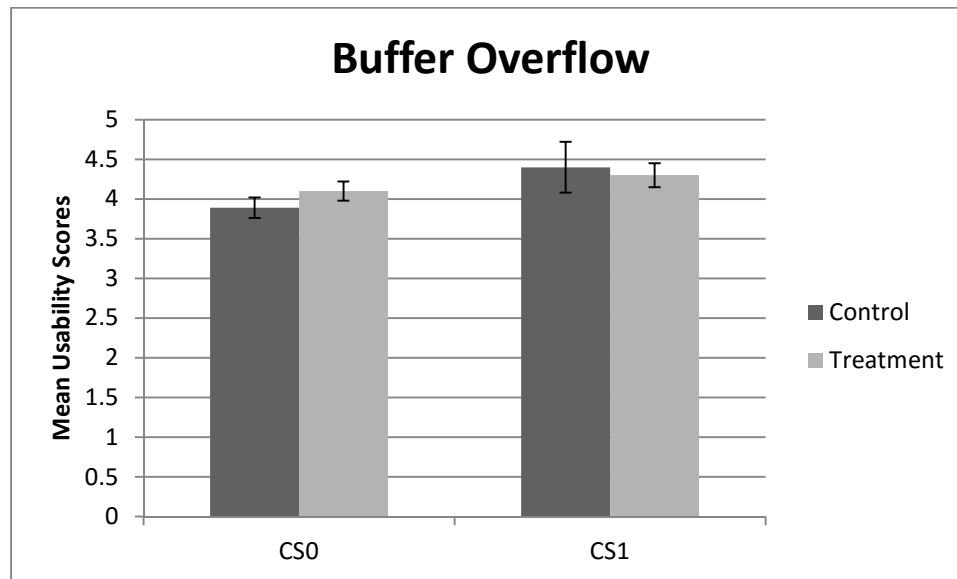


Figure 35: Mean usability scores in treatment and control group in CS0 and CS1 (Buffer Overflow)

See **Table 11** for mean scores for individual questions in integer overflow, input validation and buffer overflow modules in CS0 and CS1 (treatment and control group).

	Mean Score											
	Integer Overflow				Input Validation				Buffer Overflow			
	CS0		CS1		CS0		CS1		CS0		CS1	
	C (n = 67)	T (n = 72)	C n=33	T N=45	C (n = 27)	T (n = 52)	C n=14	T N=26	C n = 26	T n = 44	C n=9	T n=19
Q1	3.93	4.19**	3.52	4.04*	4.15	4.06	4.00	4.15	4.00	4.09	4.11	4.37
Q2	4.06	4.21	3.48	4.13*	4.11	4.17	3.86	4.27	3.96	4.27**	4.44	4.47
Q3	4.04	4.33**	3.67	4.22*	4.11	4.29	3.93	4.23	4.15	4.32	4.33	4.53
Q4	4.30	4.43	4.00	4.27	4.37	4.35	4.14	4.38	4.08	4.18	4.44	4.53
Q5	4.12	4.36	3.76	4.31*	4.15	4.27	3.64	4.42*	3.92	4.41*	4.22	4.53
Q6	4.16	4.19	3.58	4.18*	4.15	4.17	3.86	4.15	3.88	4.18**	4.44	4.26
Q7	4.28	4.42	3.94	4.38*	4.26	4.37	3.93	4.38**	4.12	4.52*	4.67	4.63
Q8	4.33	4.40	3.97	4.36**	4.30	4.38	3.79	4.23	4.08	4.50*	4.78	4.58
Q9	4.12	4.25*	3.82	4.16	4.04	4.17	3.93	4.15	4.08	4.30	4.22	4.53
Q10	4.00	4.35*	3.73	4.13**	4.11	4.17	3.57	4.15**	3.96	4.14	4.56	4.37
Q11	3.67	4.22	3.39	4.07*	3.81	4.13	3.43	4.08	3.81	4.05	4.33	3.95
Q12	3.91	4.14**	3.45	4.09*	3.93	4.15	3.36	4.12*	3.69	4.02	4.44	4.11
Q13	3.76	4.08	3.33	4.04*	3.78	3.96	3.57	4.12	3.54	4.05*	4.44	4.05
Q14	3.84	4.01	3.58	3.96**	3.63	4.04**	3.71	3.96	3.69	3.93	4.11	4.26
Q15	3.79	4.15*	3.67	4.02	3.93	4.10	3.29	4.08**	3.85	3.98	4.56	4.05
Q16	3.61	4.01*	3.55	3.82	4.04	4.02	4.00	3.92	3.73	3.98	4.44	4.00
Q17	3.91	4.15**	3.42	4.09*	4.04	4.15	3.64	4.12	3.92	3.95	4.67	4.21
Q18	3.73	3.93	3.30	3.98*	4.04	4.02	3.50	4.08**	3.62	3.95	4.11	4.00
Overall Usability Scores												
	3.97	4.21**	3.61	4.12*	4.05	4.16	3.73	4.16**	3.89	4.15	4.40	4.30
*p < 0.05 (statistically significant at 95% level), **p < 0.10 (statistically significant at 90% level), C – Control Group, T– Treatment Group												

Table 11: Mean scores for individual questions in a usability survey for integer overflow, input validation, buffer overflow in CS0 and CS1 (control and treatment)

5.4.1 Students' Comments

Student's found enhanced modules useful and provided the following comments-

1. "There should be more of this kind of modules every week for students to learn new things."
2. "Couldn't be better."
3. "Very informative and was an appropriate way to deliver information."
4. "I found it very helpful."
5. "Thank you i enjoyed it very much."

6. “It was an interesting and helpful learning experience in terms of how to properly code and the steps in which I needed to take in order to properly and fully execute the program.”
7. “Great module and less! Easy to comprehend and actually helps me learn the information.”
8. “Very interesting module.”

6. Conclusions and Discussion

The purpose of the research was to develop, implement and evaluate learning modules that could reduce content skipping, increase student engagement and learning and are usable. Using security injections @Towson cybersecurity modules, segmented and interactive learning modules were developed and delivered during fall 2014, spring 2015, fall 2015 and spring 2016 in selective CS0, CS1 and Computer Literacy courses at Computer and Information Sciences department at Towson University. The modules were laboratory-based and select modules (integer error, input validation and buffer overflow) were used to examine the research questions. Four research questions with sub-questions were set up and hypothesis was formulated based on the research instruments.

Overall, findings from this research suggest that the use of segmented and interactive learning modules reduce content skipping, increase student engagement and learning and are more usable compared to linear learning modules.

Students' mean reading scores and mean reading depth were found to be significantly ($p < 0.05$) higher using segmented – interactive modules compared to linear modules in a two group, control-group treatment-group design study, using eye-tracking conducted in fall 2015 and spring 2016.

Reading scores for each student were computed using **Buscher's** reading detection algorithm which sums all the scores assigned based on eye-movement direction (positive score in forward and negative score in backward direction) and letter spaces between the consecutive fixations, on a line of text. Reading depth is the area of text covered by the student and is computed as the ratio of number of words looked at, to total

number of words in an assigned reading.

Students read and covered more content using segmented modules (N=10, reading scores = 1970, reading depth = 0.) as compared to linear modules (N=9, reading scores = 1100, reading depth = 0.) implying reduced content skipping in segmented modules as compared to linear modules, which answers RQ1.

Students' mean engagement scores were found to be significantly ($p < 0.05$) higher using segmented – interactive modules compared to linear modules in a two group, control-group treatment-group post survey design study, conducted in fall 2014.

An eight item student engagement survey was adapted and reliability analysis was conducted with cronbach alpha of 0.74, implying good internal consistency.

Students were more engaged using segmented and interactivity modules (N = 42, engagement scores = 3.43) as compared to linear modules (N=38, engagement scores= 3.19) implying, interactive activities as an engagement factor in segmented – interactive modules, which supports the previous literature (Evans & Gibbons, 2007; Teoh & Neo, 2007) and answers RQ2.

Students' mean scores for identifying (ability to apply) software security vulnerabilities and a phishing email was found to be significantly ($p < 0.05$ for software security and $p < 0.10$ for phishing) higher using segmented – interactive modules compared to linear modules, while mean scores for general software security, secure coding awareness and phishing awareness in both the version were found to be same in a two group, control-group treatment-group pre-survey post-survey design study, conducted in spring 2015.

The pre-survey and post-survey included multiple choice questions related to

student demographics, secure coding awareness, general software security awareness and phishing awareness. In addition, 3 code segments were added to the software security post-survey to assess students' ability to apply secure coding knowledge and a sample email to identify phishing was administered during final exam.

Students performed better in identifying security vulnerabilities in code segments using segmented-interactive (N= 26, ability to apply secure coding knowledge score = 5.59) modules as compared to linear (N= 27, ability to apply secure coding knowledge score = 4.27) modules while retention of cybersecurity knowledge remained insignificant general (control (security awareness scores = 4.14 , secure coding awareness scores= 3.81), treatment (general security awareness scores = 4.0 , secure coding awareness scores= 4.21)). Similar results were found in computer literacy in phishing awareness and ability to apply phishing knowledge. This supports the previous literature that interactive systems with same content significantly improve student learning in questions that assess students' ability to apply knowledge as compared to non-interactive systems while student learning remains same in questions that assess retention of knowledge, which answers RQ3.

Students' overall mean usability scores were found to be significantly ($p < 0.05$) higher using segmented – interactive modules compared to linear modules in a two group, control-group treatment-group post survey design study, conducted in fall 2015.

An eighteen item e-learning usability survey was adapted and reliability analysis was conducted with cronbach alpha of 0.97, implying strong internal consistency.

Students found segmented and interactivity modules (N = 332, overall usability scores = 4.16) more usable as compared to linear modules (N=206, overall usability

scores=3.88) implying that students found modules effective, efficient and satisfying. In addition, supports theory from literature that in order to improve learning outcomes in an e-learning system, the system must be usability. This answers RQ4.

Overall, we were successful in developing and implementing in learning modules that reduce content skipping, increase student engagement and learning and are usable. Statistically significant changes were demonstrated in students' reading scores and reading depth; students' ability to apply knowledge; student engagement and overall usability of segmented-interactive modules.

Future work includes reassessing results for RQ1, RQ2, RQ3 and RQ4 with larger sample size across multiple courses and areas. In addition, improving eye-tracking procedure to collect the data with high accuracy and precision for usability analysis of the segmented and interactive modules.

Appendices

Appendix A – Institutional Review Board Documents

From: IRB

Sent: Thursday, October 22, 2015 12:57 PM

To: Kaza, Siddharth <SKaza@towson.edu>; Taylor, Blair <btaylor@towson.edu>

Subject: IRB protocol 09-0xii - modification # 2

Hi Sidd and Blair,

I heard from the reviewer and the modifications you submitted will be approved, the protocol will remain as exempt. A formal letter will go out to you tomorrow in campus mail, but you can accept this email as approval at this time.



TOWSON
UNIVERSITY

Amy L. Taylor · Assistant Vice President for Research

Office of Sponsored Programs & Research Academic Affairs

Towson University · 8000 York Road · Towson, Maryland, 21252-0001

t. [410-704-4931](tel:410-704-4931) · f. [410-704-4494](tel:410-704-4494)

Appendix B – Assessment for Student Learning

APPENDICES

APPENDIX 1: Pre-Survey CS0, CS1

Dear Participant,

The purpose of this experiment is to evaluate student knowledge of security concepts and principles. This is part of an NSF-funded research program aimed at analyzing the effectiveness of infusing security principles into undergraduate classes. This research is being funded by a grant from the National Science Foundation.

Participation in this study is voluntary. If you choose to participate in this project, you will be asked to complete a short survey. It is not necessary to answer every question, and you may discontinue your participation in the survey at any time. Your decision whether or not to participate in the survey or to withdraw from the project at any time will in no way affect your class standing, or if you are an athlete, your status as an athlete.

If you have any questions about the project, you may contact Blair Taylor/Siddharth Kaza (securityinjections@towson.edu) or Towson University's Institutional Review Board for the Protection of Human Participants, irb@towson.edu at (410) 704-2236. A copy of the survey results, reported in aggregate form, will be available to you upon request.

Thank you for your time and willingness to participate in this survey.

Sincerely,

Blair Taylor / Siddharth Kaza
Department of Computer and Information Sciences
Principal Investigator

Demographics

1. What is your gender?
 - a) Male
 - b) Female
2. What is your age ?
 - a) 20 years or younger
 - b) 21-25 years
 - c) 26-30 years
 - d) 31 years or older
3. Which ethnic group best describes you ?
 - a) White
 - b) Black
 - c) Hispanic
 - d) Asian
 - e) Other
4. What is your current student standing?
 - a) Freshman
 - b) Sophomore
 - c) Junior
 - d) Senior
 - e) Other
5. What is your major?
 - a) Information Systems or Computer Information Systems
 - b) Computer Science
 - c) Computer Technology or Information Technology
 - d) Mathematics
 - e) Undecided
 - f) Other

Cyber Security Interest

6. Based on your level of agreement on the scale of 7, do you code securely?
 - a) Strongly Agree
 - b) Agree
 - c) Agree Somewhat
 - d) Undecided
 - e) Disagree Somewhat
 - f) Disagree
 - g) Strongly Disagree
7. How likely is it that you will major in computer security track

- Not at all likely 1 2 3 4 5 6 7 Extremely Likely
8. How likely is it that you read magazine or newspaper articles related to secure coding
- Not at all likely 1 2 3 4 5 6 7 Extremely Likely
9. How likely is it that you participate in a club or organization related to secure coding
- Not at all likely 1 2 3 4 5 6 7 Extremely Likely
10. How important is learning secure coding principles for you
- Not at all important 1 2 3 4 5 6 7 Extremely important
11. How important is for you to learn new ways of coding securely
- Not at all important 1 2 3 4 5 6 7 Extremely important

Cyber Security Awareness

12. What are the possible consequences of insufficient computer security?
- a) I may have files deleted from my computer
 - b) I may have personal communications exposed
 - c) I may have my network connection cut off
 - d) All of the above
 - e) Unsure
13. Integer Overflow occurs?
- a) when a number exceeds the largest possible value
 - b) when the run-time stack runs out of storage
 - c) when the bounds of an array are exceeded
 - d) Unsure
14. Integer Overflow is caused by?
- a) Virus
 - b) Unchecked input or an operation such as multiplication or exponentiation
 - c) an array overflow
 - d) Unsure
15. Phishing is?
- a) a program that monitors your internet activity
 - b) hacking
 - c) fraudulent email asking for personal information that can be used in identity theft
 - d) Unsure
16. The conversion of data into a ciphertext that cannot be easily understood by unauthorized people is known as:
- a) brute force hacking

- b) tunneling
 - c) encryption
 - d) ciphertext feedback
 - e) cloaking
 - f) unsure
17. Security Software and Software Security are the same:
- a) True
 - b) False
 - c) unsure
18. When developing secure systems, where does security fit in ?:
- a) After design is complete
 - b) During testing
 - c) Before implementation
 - d) After implementation
 - e) At all phases of development
 - f) Unsure
19. Software security vulnerabilities are the result of software bugs and flaws:
- a) True
 - b) False
 - c) Unsure
20. Which programming mistake is one of the major vulnerabilities in today's applications ?:
- a) Undocumented code
 - b) Buffer overflow
 - c) Weak passwords
 - d) Compiler bugs
 - e) Unsure
21. A set of related programs, usually located at a network gateway server, that protects the resources of a private network from other networks, is known as a:
- a) firewall
 - b) sandbox
 - c) rootkit
 - d) password cracker
 - e) general protection fault
 - f) Unsure
22. Which of the following should your well-designed program do before processing user input ?:
- a) Verify that the data is of the correct type (number, string, etc).

- b) Verify that the data value is appropriate (ages should not be negative numbers, etc.)
- c) Examine the data to make sure that there are no suspicious values that might indicate attempts at exploiting security holes
- d) All of the above
- e) Unsure

APPENDIX 2: Post survey CS0, CS1, CS2

Dear Participant,

The purpose of this experiment is to evaluate student knowledge of security concepts and principles. This is part of an NSF-funded research program aimed at analyzing the effectiveness of infusing security principles into undergraduate classes. This research is being funded by a grant from the National Science Foundation.

Participation in this study is voluntary. If you choose to participate in this project, you will be asked to complete a short survey. It is not necessary to answer every question, and you may discontinue your participation in the survey at any time. Your decision whether or not to participate in the survey or to withdraw from the project at any time will in no way affect your class standing, or if you are an athlete, your status as an athlete.

If you have any questions about the project, you may contact Blair Taylor/Siddharth Kaza (securityinjections@towson.edu) or Towson University's Institutional Review Board for the Protection of Human Participants, irb@towson.edu at [\(410\) 704-2236](tel:4107042236). A copy of the survey results, reported in aggregate form, will be available to you upon request.

Thank you for your time and willingness to participate in this survey.

Sincerely,

Blair Taylor / Siddharth Kaza
Department of Computer and Information Sciences
Principal Investigator

Demographics

1. What is your gender?
 - a) Male
 - b) Female
2. What is your age ?
 - a) 20 years or younger
 - b) 21-25 years
 - c) 26-30 years
 - d) 31 years or older
3. Which ethnic group best describes you ?
 - a) White
 - b) Black
 - c) Hispanic
 - d) Asian
 - e) Other
4. What is your current student standing?
 - a) Freshman
 - b) Sophomore
 - c) Junior
 - d) Senior
 - e) Other
5. What is your major?
 - a) Information Systems or Computer Information Systems
 - b) Computer Science
 - c) Computer Technology or Information Technology
 - d) Mathematics
 - e) Undecided
 - f) Other

User-System engagement

Based on your level of agreement on the scale of 5, answer the following :

6. I felt deeply engrossed in completing the security injection modules using this web-based platform.
 - a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
7. I get so involved while completing security injection modules using this web-based platform that I forget everything.

- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
8. While completing the security injection modules using this web-based platform, I tend to block out conversations with others around me.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
9. The security injection modules presented on this platform hold my attention.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
10. Using this web-based platform excited my curiosity to learn cyber security principles.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
11. Time seemed to go by very quickly when I use this web-based platform for completing security injection module.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
12. The screen layout of this web-based platform for security injection modules was visually pleasing.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree

13. Using this web-based platform for security injection modules was mentally taxing.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
14. Using web-based platform for completing security injection modules was attractive.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree

Student-Interest in cyber security

15. Based on your level of agreement on the scale of 7, do you code securely?
- a) Strongly Agree
 - b) Agree
 - c) Agree Somewhat
 - d) Undecided
 - e) Disagree Somewhat
 - f) Disagree
 - g) Strongly Disagree
16. How likely is it that you will major in computer security track
- | | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|------------------|
| Not at all likely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Extremely Likely |
|-------------------|---|---|---|---|---|---|---|------------------|
17. How likely is it that you read magazine or newspaper articles related to cyber security
- | | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|------------------|
| Not at all likely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Extremely Likely |
|-------------------|---|---|---|---|---|---|---|------------------|
18. How likely is it that you participate in a club or organization related to cyber security
- | | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|------------------|
| Not at all likely | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Extremely Likely |
|-------------------|---|---|---|---|---|---|---|------------------|
19. How important is learning secure coding principles for you
- | | | | | | | | | |
|----------------------|---|---|---|---|---|---|---|---------------------|
| Not at all important | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Extremely important |
|----------------------|---|---|---|---|---|---|---|---------------------|
20. How important is for you to learn new ways of coding securely
- | | | | | | | | | |
|----------------------|---|---|---|---|---|---|---|---------------------|
| Not at all important | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Extremely important |
|----------------------|---|---|---|---|---|---|---|---------------------|

Students' Cyber Security Awareness

21. What are the possible consequences of insufficient computer security?
- a) I may have files deleted from my computer

- b) I may have personal communications exposed
 - c) I may have my network connection cut off
 - d) All of the above
 - e) Unsure
22. Integer Overflow occurs?
- a) when a number exceeds the largest possible value
 - b) when the run-time stack runs out of storage
 - c) when the bounds of an array are exceeded
 - d) Unsure
23. Integer Overflow is caused by?
- a) Virus
 - b) Unchecked input or an operation such as multiplication or exponentiation
 - c) an array overflow
 - d) Unsure
24. Phishing is?
- a) a program that monitors your internet activity
 - b) hacking
 - c) fraudulent email asking for personal information that can be used in identity theft
 - d) Unsure
25. The conversion of data into a ciphertext that cannot be easily understood by unauthorized people is known as:
- a) brute force hacking
 - b) tunneling
 - c) encryption
 - d) ciphertext feedback
 - e) cloaking
 - f) unsure
26. Security Software and Software Security are the same:
- a) True
 - b) False
 - c) unsure
27. When developing secure systems, where does security fit in?:
- a) After design is complete
 - b) During testing
 - c) Before implementation
 - d) After implementation
 - e) At all phases of development
 - f) Unsure

28. Software security vulnerabilities are the result of software bugs and flaws:
- a) True
 - b) False
 - c) Unsure
29. Which programming mistake is one of the major vulnerabilities in today's applications?:
- a) Undocumented code
 - b) Buffer overflow
 - c) Weak passwords
 - d) Compiler bugs
 - e) Unsure
30. A set of related programs, usually located at a network gateway server, that protects the resources of a private network from other networks, is known as a:
- a) firewall
 - b) sandbox
 - c) rootkit
 - d) password cracker
 - e) general protection fault
 - f) Unsure
31. Which of the following should your well-designed program do before processing user input ?:
- a) Verify that the data is of the correct type (number, string, etc).
 - b) Verify that the data value is appropriate (ages should not be negative numbers, etc.)
 - c) Examine the data to make sure that there are no suspicious values that might indicate attempts at exploiting security holes
 - d) All of the above
 - e) Unsure
32. Your code is completely secure if...
- a) It is written in Java.
 - b) It executes correctly for all valid input
 - c) You have used a firewall and anti-virus software
 - d) There is no such thing as completely secure code
 - e) Unsure
33. Which of the following is an example of strong password?
- a) Passcode
 - b) J*p2le04
 - c) Your real name, user name or company name
 - d) Unsure

34. Invalid input can come from?

- a) Keyboard
- b) Network
- c) Disk drive
- d) All of the above
- e) Unsure

35. Identify the potential security issues in the following code segment, select all that apply:

```
float price;  
float totalPrice;  
cout << "Enter Price" << endl;  
cin >> price;  
totalPrice = price + price*.06;
```

- A) Integer Overflow or Underflow
- B) Input Validation vulnerabilities
- C) Buffer Overflow

36. Identify the potential security issues in the following code segment, select all that apply:

```
int calc (int i, int j) //assume i < INT_MAX and j < INT_MAX  
{  
    int result = i * j;  
    return result;  
}
```

- A) Integer Overflow or Underflow
- B) Input Validation vulnerabilities
- C) Buffer Overflow

37. Identify the potential security issues in the following code segment, select all that apply:

```
void input(float temperatures[], int n) // assume n < INT_MAX  
{  
    for (int i = 0; i < n; i = i + 1)  
    {  
        cout << temperatures[i] << endl;  
    }  
}
```

- A) Integer Overflow or Underflow

- B) Input Validation vulnerabilities
- C) Buffer Overflow

APPENDIX 3: Pre Survey Computer Literacy

Dear Participant,

The purpose of this experiment is to evaluate student knowledge of security concepts and principles. This is part of an NSF-funded research program aimed at analysing the effectiveness of infusing security principles into undergraduate classes. This research is being funded by a grant from the National Science Foundation.

Participation in this study is voluntary. If you choose to participate in this project, you will be asked to complete a short survey. It is not necessary to answer every question, and you may discontinue your participation in the survey at any time. Your decision whether or not to participate in the survey or to withdraw from the project at any time will in no way affect your class standing or, if you are an athlete, your status as an athlete.

If you have any questions about the project, you may contact Blair Taylor/Siddharth Kaza (securityinjections@towson.edu) or Towson University's Institutional Review Board for the Protection of Human Participants, irb@towson.edu at (410) 704-2236. A copy of the survey results, reported in aggregate form, will be available to you upon request.

Thank you for your time and willingness to participate in this survey.

Sincerely,

Blair Taylor / Siddharth Kaza
Department of Computer and Information Sciences
Principal Investigator

Enter your student ID

Demographics

1. What is your gender?
 - a) Male
 - b) Female
2. What is your age?
 - a) 20 years or younger
 - b) 21-25 years
 - c) 26-30 years
 - d) 31 years or older
3. Which ethnic group best describes you?
 - a) White
 - b) Black
 - c) Hispanic
 - d) Asian
 - e) Multi-racial
 - f) Other
4. What is your current student standing?
 - a) Freshman
 - b) Sophomore
 - c) Junior
 - d) Senior
 - e) Other
5. What is your major?
 - a) Information Systems or Computer Information Systems
 - b) Computer Science
 - c) Computer Technology or Information Technology
 - d) Mathematics
 - e) Undecided
 - f) Other
6. What is the name of the course?
 - a) CS0
 - b) CS1
 - c) CS2
 - d) Computer Literacy
 - e) Database Management
 - f) Other

Computer Security Awareness

7. As this survey will study how your understanding of computer security issues changes from the beginning of the semester to the end of the semester, we will need you to select a secret code that you will enter each time you take this survey. In order to keep your responses anonymous, this code should be not be known to any of the teaching staff. To find your security code, use the following procedure:
1. Multiply the day of your birth by 10. Thus, if you were born on the 13th, use 130.
 2. Add that number to the last 3 digits of your phone number. Thus, if your phone number is 555 1212, and you were born on the 30th, you would have $212+130=342$
 3. If the sum is more than 1000, subtract 1000 from it. For example, if your sum was 1192, subtract 1000 to get 192
 4. The resulting number is your code number

Please enter your code number: _____ [numeric textbox]

8. What are the possible consequences of insufficient computer security?
- a) I may have files deleted from my computer
 - b) I may have personal communications exposed
 - c) I may have my network connection cut off
 - d) My computer may be used to commit a crime
 - e) All of the above
 - f) Unsure
9. Phishing is:
- a) a program that monitors your Internet activity
 - b) hacking
 - c) fraudulent email asking for personal information that can be used in identity theft
 - d) Unsure
10. A set of related programs, usually located at a network gateway server, that protects the resources of a private network from other networks, is known as a:
- a) firewall
 - b) sandbox
 - c) rootkit

- d) password cracker
 - e) general protection fault
 - f) Unsure
11. Who is it safe to tell your password to?
- a) Ebay, if they send you an email first
 - b) Your best friend, in case you forget it
 - c) A colleague who needs to send you an urgent email
 - d) You should never disclose your password to anyone
 - e) Unsure
12. Encryption is a special technique employed only by agencies with highly sensitive data such as the FBI or CIA.
- a) True
 - b) False
 - c) Unsure

13. Consider the following email:

From: support@citibank.com
Subject: Verify your E-mail with Citibank

Dear Citibank Member,

This email was sent by the Citibank server to verify your email address. You must complete this process by clicking on the link below and entering in the small window your Citibank ATM/Debit Card number and PIN that you use on ATM.

This is done for your protection - because some of our members no longer have access to their email addresses and we must verify it.

To verify your E-mail address and access your bank account, click on the link below:

https://web.da-us.citibank.com/signin/citifi/scripts/email_verify.jsp

Thank you for using Citibank

Is the above email:

- a) Legitimate
- b) Fraudulent
- c) Unsure

14. I can be held responsible for what others do while using my password.
- a) True
 - b) False
 - c) Unsure
15. The conversion of data into a code that cannot be easily understood by unauthorized people is known as:
- a) brute force hacking
 - b) tunneling
 - c) encryption
 - d) cloaking
 - e) unsure
16. You should ensure that companies and other organization with whom you do business encrypt your personal data, such as credit card numbers and social security numbers, before they are stored or transmitted over a network.
- a) True
 - b) False
 - c) unsure
17. The following are characteristics of suspicious email
- a) Grammatical or spelling errors in the e-mail
 - b) the e-mail contain an air of urgency or a need to respond immediately
 - c) a and b
 - d) comes from a trusted user
 - e) unsure
18. Using letters from a memorable phrase is a recommended way to construct a password.
- a) True
 - b) False
 - c) Unsure
19. Never give out personal information upon an email request.
- a) True
 - b) False
 - c) Unsure

20. Which of the following is an example of a strong password?

- a) Password
- b) J*p2le04>F
- c) Your real name, user name or company named
- d) D. A1*
- e) Unsure

21. Encrypting your personal files requires purchasing special software.

- a) True
- b) False
- c) Unsure

22. How interested are you in security?

- a) Extremely interested
- b) Very interested
- c) Somewhat interested
- d) Slightly interested
- e) Not at all interested

23. How important do you think security knowledge is to your future career?

- a) Extremely important
- b) Very important
- c) Somewhat important
- d) Slightly important
- e) Not at all important

APPENDIX 4: Post Survey Computer Literacy

Dear Participant,

The purpose of this experiment is to evaluate student knowledge of security concepts and principles. This is part of an NSF-funded research program aimed at analysing the effectiveness of infusing security principles into undergraduate classes. This research is being funded by a grant from the National Science Foundation.

Participation in this study is voluntary. If you choose to participate in this project, you will be asked to complete a short survey. It is not necessary to answer every question, and you may discontinue your participation in the survey at any time. Your decision whether or not to participate in the survey or to withdraw from the project at any time will in no way affect your class standing, or if you are an athlete, your status as an athlete.

If you have any questions about the project, you may contact Blair Taylor/Siddharth Kaza (securityinjections@towson.edu) or Towson University's Institutional Review Board for the Protection of Human Participants, irb@towson.edu at (410) 704-2236. A copy of the survey results, reported in aggregate form, will be available to you upon request.

Thank you for your time and willingness to participate in this survey.

Sincerely,

Blair Taylor / Siddharth Kaza
Department of Computer and Information Sciences
Principal Investigator

Enter your student ID

Demographics

1. What is your gender?
 - a) Male
 - b) Female
2. What is your age?
 - a) 20 years or younger
 - b) 21-25 years
 - c) 26-30 years
 - d) 31 years or older
3. Which ethnic group best describes you?
 - a) White
 - b) Black
 - c) Hispanic
 - d) Asian
 - e) Other
4. What is your current student standing?
 - a) Freshman
 - b) Sophomore
 - c) Junior
 - d) Senior
 - e) Other
5. What is your major?
 - a) Information Systems or Computer Information Systems
 - b) Computer Science
 - c) Computer Technology or Information Technology
 - d) Mathematics
 - e) Undecided
 - f) Other
6. What is the name of the course?
 - a) CS0
 - b) CS1
 - c) CS2
 - d) Computer Literacy
 - e) Database Management
 - f) Other

Computer Security Awareness

7. As this survey will study how your understanding of computer security issues changes from the beginning of the semester to the end of the semester, we will need you to select a secret code that you will enter each time you take this survey. In order to keep your responses anonymous, this code should be not be known to any of the teaching staff. To find your security code, use the following procedure:
 1. Multiply the day of your birth by 10. Thus, if you were born on the 13th, use 130.
 2. Add that number to the last 3 digits of your phone number. Thus, if your phone number is 555 1212, and you were born on the 30th, you would have $212+130=342$
 3. If the sum is more than 1000, subtract 1000 from it. For example, if your sum was 1192, subtract 1000 to get 192
 4. The resulting number is your code number

Please enter your code number: _____

8. What are the possible consequences of insufficient computer security?
 - a) I may have files deleted from my computer
 - b) I may have personal communications exposed
 - c) I may have my network connection cut off
 - d) My computer may be used to commit a crime
 - e) All of the above
 - f) Unsure
9. Phishing is:
 - a) a program that monitors your internet activity
 - b) hacking
 - c) fraudulent email asking for personal information that can be used in identity theft
 - d) Unsure
10. A set of related programs, usually located at a network gateway server, that protects the resources of a private network from other networks, is known as a:
 - a) firewall
 - b) sandbox
 - c) rootkit

- d) password cracker
 - e) general protection fault
 - f) Unsure
11. Who is it safe to tell your password to?
- a) Ebay, if they send you an email first
 - b) Your best friend, in case you forget it
 - c) A colleague who needs to send you an urgent email
 - d) You should never disclose your password to anyone
 - e) Unsure
12. Encryption is a special technique employed only by agencies with highly sensitive data such as the FBI or CIA.
- a) True
 - b) False
 - c) Unsure

13. Consider the following email:

From: support@citibank.com
Subject: Verify your E-mail with Citibank

Dear Citibank Member,

This email was sent by the Citibank server to verify your email address. You must complete this process by clicking on the link below and entering in the small window your Citibank ATM/Debit Card number and PIN that you use on ATM.

This is done for your protection - because some of our members no longer have access to their email addresses and we must verify it.

To verify your E-mail address and access your bank account, click on the link below:

https://web.da-us.citibank.com/signin/citifi/scripts/email_verify.jsp

Thank you for using Citibank

Is the above email:

- a) Legitimate
- b) Fradulent
- c) Unsure

14. I can be held responsible for what others do while using my password.
- a) True
 - b) False
 - c) Unsure
15. The conversion of data into a code that cannot be easily understood by unauthorized people is known as:
- a) brute force hacking
 - b) tunneling
 - c) encryption
 - d) cloaking
 - e) unsure
16. You should ensure that companies and other organization with whom you do business encrypt your personal data, such as credit card numbers and social security numbers, before they are stored or transmitted over a network.
- d) True
 - e) False
 - f) unsure
17. The following are characteristics of suspicious email
- a) Grammatical or spelling errors in the e-mail
 - b) the e-mail contain an air of urgency or a need to respond immediately
 - c) a and b
 - d) comes from a trusted user
 - e) unsure
18. Using letters from a memorable phrase is a recommended way to construct a password.
- a) True
 - b) False
 - c) Unsure
19. Never give out personal information upon an email request
- a) True
 - b) False
 - c) Unsure

20. Which of the following is an example of a strong password?

- a) Password
- b) J*p2le04>F
- c) Your real name, user name or company named
- d) D. A1*
- e) Unsure

21. Encrypting your personal files requires purchasing special software.

- a) True
- b) False
- c) Unsure

22. How interested are you in security?

- a) Extremely interested
- b) Very interested
- c) Somewhat interested
- d) Slightly interested
- e) Not at all interested

23. How important do you think security knowledge is to your future career?

- a) Extremely important
- b) Very important
- c) Somewhat important
- d) Slightly important
- e) Not at all important

User-System engagement

Please indicate your level of agreement with each of the following statements:

24. I felt deeply engrossed in completing the security injection modules using this web-based platform.

- a) Strongly Agree
- b) Agree
- c) Neutral
- d) Disagree
- e) Strongly Disagree

25. I get so involved while completing security injection modules using this web-based platform that I forget everything.

- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
26. While completing the security injection modules using this web-based platform, I tend to block out conversations with others around me.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
27. The security injection modules presented on this platform hold my attention.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
28. Using this web-based platform excited my curiosity to learn cyber security principles.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
29. Time seemed to go by very quickly when I use this web-based platform for completing security injection module.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree
30. The screen layout of this web-based platform for security injection modules was visually pleasing.
- a) Strongly Agree
 - b) Agree
 - c) Neutral
 - d) Disagree
 - e) Strongly Disagree

31. Using this web-based platform for security injection modules was mentally taxing.

- a) Strongly Agree
- b) Agree
- c) Neutral
- d) Disagree
- e) Strongly Disagree

32. Using web-based platform for completing security injection modules was attractive.

- a) Strongly Agree
- b) Agree
- c) Neutral
- d) Disagree
- e) Strongly Disagree

Do you have any additional comments?

Yes (please explain)

No

(Ability to apply phishing knowledge)

33. Consider the following email:

From: Help Desk <online2793774@telkomsa.net>

Date: June 20, 2014 at 7:57:55 AM PDT

To: info@cs.stanford.edu

Subject: update

It had been detected that your cs-stanford-edu email account. Mail delivery system had been affected with virus. Your email account had been sending virus included with your mail to recipient's account and as such a threat to our database. You'll need to update the settings on your cs-stanford-edu email account by clicking on this link:

http://forms.logiforms.com/formdata/user_forms/66949_9366478/321793

From

CS. Standford

ITS Helpdesk

33a. Is the above email, **legitimate** or **fraudulent**?

33b. What makes you decide, the above email is **legitimate** or **fraudulent**? Discuss elaborately.

APPENDIX 5: Module Usability Survey

Dear Participant,

The purpose of this experiment is to evaluate usability of security injections web platform. This is part of an NSF-funded research program aimed at analysing the efficiency, effectiveness and student satisfaction of security injections web-based platform. This research is being funded by a grant from the National Science Foundation.

Participation in this study is voluntary. If you choose to participate in this project, you will be asked to complete a survey. It is not necessary to answer every question, and you may discontinue your participation in the survey at any time. Your decision whether or not to participate in the survey or to withdraw from the project at any time will in no way affect your class standing, or if you are an athlete, your status as an athlete.

If you have any questions about the project, you may contact Blair Taylor/Siddharth Kaza (securityinjections@towson.edu) or Towson University's Institutional Review Board for the Protection of Human Participants, irb@towson.edu at [\(410\) 704-2236](tel:4107042236). A copy of the survey results, reported in aggregate form, will be available to you upon request.

Thank you for your time and willingness to participate in this survey.

Sincerely,

Blair Taylor / Siddharth Kaza
Department of Computer and Information Sciences
Principal Investigator

Enter your student ID

Demographics

1. What is your gender?
 - a) Male
 - b) Female
2. What is your age?
 - a) 20 years or younger
 - b) 21-25 years
 - c) 26-30 years
 - d) 31 years or older
3. Which ethnic group best describes you?
 - a) White
 - b) Black
 - c) Hispanic
 - d) Asian
 - e) Other
4. What is your current student standing?
 - a) Freshman
 - b) Sophomore
 - c) Junior
 - d) Senior
 - e) Other
5. What is your major?
 - a) Information Systems or Computer Information Systems
 - b) Computer Science
 - c) Computer Technology or Information Technology
 - d) Mathematics
 - e) Undecided
 - f) Other
6. What is the name of the course?
 - a) CS0
 - b) CS1
 - c) CS2
 - d) Computer Literacy
 - e) Database Management
 - f) Other
7. What security injections version did you use?

- a) Non-interactive (Security Injections 1.0)
- b) Interactive (Security Injections 2.0)

8. What security injections module did you just completed ?

- a) Integer Overflow
- b) Input Validation
- c) Buffer Overflow
- d) Secure Development Life Cycle
- e) Phishing
- f) Cryptography
- g) Passwords
- h) Social Networking Security

Please indicate your level of agreement with the following statements

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Instructions to use the module were clear.					
I found it easy to navigate around the module.					
The module is easy to launch					
The fonts, colors, and sizes are consistent throughout the module					
The module maintains an appropriate level of consistency in its design from one part/section of the module to another.					
I found the interface clear, structured and appealing.					
Text and graphics are legible.					
Fonts (style, color, saturation) are easy to read.					
The module does not provide too many long sections of text to read without meaningful interactions					
The module engaged me in interactive tasks that are closely aligned with the					

learning goals and objectives					
The module used interactive activities to gain the attention, sustain the interest, and maintain my motivation.					
Questions in the module enhanced my understanding of cybersecurity ideas and concepts.					
Security checklist in the module enhanced my understanding of cybersecurity ideas and concepts.					
Feedback on activities is clear and helpful in learning.					
The module provides guidance and support to complete individual sections including learning activities					
I was able to complete the module quickly					
I was able to effectively complete the module					
It was simple to use the module					
I was satisfied with the module					
Additional Comments:					

List of References

- Al-Samarraie, H., Teo, T., & Abbas, M. (2013). Can structured representation enhance students' thinking skills for better understanding of E-learning content? *Computers & Education*, 69, 463–473. doi:10.1016/j.compedu.2013.07.038
- Altun, A. (2000). Patterns in Cognitive Processes and Strategies in Hypertext Reading: A Case Study of Two Experienced Computer Users. *Journal of Educational Multimedia and Hypermedia*, 9(1), 35–55. Retrieved from <http://www.editlib.org/p/8076/>
- Ardito, C., Costabile, M. F., Marsico, M. De, Lanzilotti, R., Levialdi, S., Roselli, T., & Rossano, V. (2005). An approach to usability evaluation of e-learning applications. *Universal Access in the Information Society*, 4(3), 270–283. doi:10.1007/s10209-005-0008-6
- Ariasi, N., & Mason, L. (2010). Uncovering the effect of text structure in learning from a science text: An eye-tracking study. *Instructional Science*, 39(5), 581–601. doi:10.1007/s11251-010-9142-5
- Atterer, R., Wnuk, M., & Schmidt, A. (2006). Knowing the user's every move. In *Proceedings of the 15th international conference on World Wide Web - WWW '06* (p. 203). New York, New York, USA: ACM Press. doi:10.1145/1135777.1135811
- Avouris, N. M., Dimitracopoulou, A., Daskalaki, S., & Tselios, N. K. (2001). Evaluation of Distance-Learning Environments: Impact of Usability on Student Performance. *International Journal of Educational Telecommunications*, 7(4), 355–378. Retrieved from <http://www.editlib.org/p/9213/>
- Azevedo, R., & Bernard, R. M. (1994). The Effects of Computer-Presented Feedback on Learning from Computer-Based Instruction: A Meta-Analysis. Retrieved from <http://eric.ed.gov/?id=ED385235>
- Bednarik, R., Busjahn, T., Schulte, C., & Tamm, S. (n.d.). Eye movements in programming: models to data : proceedings of the Third International Workshop. University of Eastern Finland. Retrieved from http://epublications.uef.fi/pub/urn_isbn_978-952-61-2040-9/index_en.html
- Beymer, D., Russell, D. M., & Orton, P. Z. (2005). *Human-Computer Interaction - INTERACT 2005*. (M. F. Costabile & F. Paternò, Eds.) (Vol. 3585). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/11555261
- Biedert, R., Hees, J., Dengel, A., & Buscher, G. (2012). A robust realtime reading-skimming classifier. In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '12* (p. 123). New York, New York, USA: ACM Press. doi:10.1145/2168556.2168575

- Brinck, T., Gergle, D., & Wood, S. D. (2001). Usability for the Web: Designing Web Sites that Work. Retrieved from <http://dl.acm.org/citation.cfm?id=384232>
- Buscher, G., Dengel, A., & van Elst, L. (2008). Eye movements as implicit relevance feedback. In *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08* (p. 2991). New York, New York, USA: ACM Press. doi:10.1145/1358628.1358796
- Busjahn, T., Bednarik, R., Begel, A., Crosby, M., Paterson, J. H., Schulte, C., ... Tamm, S. (2015). Eye Movements in Code Reading: Relaxing the Linear Order. In *2015 IEEE 23rd International Conference on Program Comprehension* (pp. 255–265). IEEE. doi:10.1109/ICPC.2015.36
- Busjahn, T., Shchekotova, G., Antropova, M., Schulte, C., Sharif, B., Begel, A., ... Ihantola, P. (2014). Eye tracking in computing education. In *Proceedings of the tenth annual conference on International computing education research - ICER '14* (pp. 3–10). New York, New York, USA: ACM Press. doi:10.1145/2632320.2632344
- Campbell, C. S., & Maglio, P. P. (2001). A robust algorithm for reading detection. In *Proceedings of the 2001 workshop on Perceptive user interfaces - PUI '01* (p. 1). New York, New York, USA: ACM Press. doi:10.1145/971478.971503
- Cangoz, B., & Altun, A. (2012). The Effects of Hypertext Structure, Presentation, and Instruction Types on Perceived Disorientation and Recall Performances. *CONTEMPORARY EDUCATIONAL TECHNOLOGY*, 3(2), 81–98. Retrieved from <http://www.cedtech.net/articles/32/321.pdf>
- Chin, C., & Brown, D. E. (2000). Learning in Science: A Comparison of Deep and Surface Approaches. *Journal of Research in Science Teaching*, 37(2), 109–138. doi:10.1002/(SICI)1098-2736(200002)37:2<109::AID-TEA3>3.0.CO;2-7
- Chuang, H.-H., & Liu, H.-C. (2011). Effects of Different Multimedia Presentations on Viewers' Information-Processing Activities Measured by Eye-Tracking Technology. *Journal of Science Education and Technology*, 21(2), 276–286. doi:10.1007/s10956-011-9316-1
- Clark, R. C., & Mayer, R. E. (2011). *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning* (Google eBook). John Wiley & Sons. Retrieved from <http://books.google.com/books?hl=en&lr=&id=twoLz3jlkRgC&pgis=1>
- Cohen, A. L. (2013). Software for the automatic correction of recorded eye fixation locations in reading experiments. *Behavior Research Methods*, 45(3), 679–83. doi:10.3758/s13428-012-0280-3
- CSIS. (2010). *A Human Capital Crisis in Cybersecurity*. Washington DC. Retrieved from

- http://csis.org/files/publication/100720_Lewis_HumanCapital_WEB_BlkwhteVersion.pdf
- DeStefano, D., & LeFevre, J.-A. (2007). Cognitive load in hypertext reading: A review. *Computers in Human Behavior*, 23(3), 1616–1641. doi:10.1016/j.chb.2005.08.012
- Duggan, G. B., & Payne, S. J. (2011). Skim Reading by Satisficing: Evidence from Eye Tracking. In *CHI 2011*. Vancouver, BC, Canada. Retrieved from http://delivery.acm.org/10.1145/1980000/1979114/p1141-duggan.pdf?ip=136.160.164.67&id=1979114&acc=ACTIVE_SERVICE&key=5F8E7AA76238C9EB.520321CBDF045ED7.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=441326838&CFTOKEN=61768444&__acm__=1397931691_eb42f911a72e1c8efbf49fdafafb86e2
- Dyson, M., & Haselgrove, M. (2000). The effects of reading speed and reading patterns on the understanding of text read from screen. *Journal of Research in Reading*, 23(2), 210–223. doi:10.1111/1467-9817.00115
- Entwistle, N. (2000). Promoting deep learning through teaching and assessment: conceptual frameworks and educational contexts. In *TLRP Conference*. Retrieved from <http://www.tlrp.org/pub/acadpub/Entwistle2000.pdf>
- Evans, C., & Gibbons, N. J. (2007). The interactivity effect in multimedia learning. *Computers & Education*, 49(4), 1147–1160. doi:10.1016/j.compedu.2006.01.008
- Floyd, K. S., Harrington, S. J., & Santiago, J. (2009). The Effect of Engagement and Perceived Course Value on Deep and Surface Learning Strategies. *Informing Science: The International Journal of an Emerging Transdiscipline*, 12. Retrieved from <http://inform.nu/Articles/Vol12/ISJv12p181-190Floyd530.pdf>
- Hattie, J. (2013). *Visible Learning: A Synthesis of Over 800 Meta-Analyses Relating to Achievement*. Routledge. Retrieved from <https://books.google.com/books?hl=en&lr=&id=ZO8jmUjQbs0C&pgis=1>
- Hegarty, B. (2005). *Evaluation Report Essay Writing - Usability of Prototype Results for Information Literacy e-Learning Modules - Reusable and Portable across a College of Education, a Polytechnic and a University*. Retrieved from <http://oil.otago.ac.nz/oil/>
- Hessler, K. L., & Henderson, A. M. (2013). Interactive Learning Research: Application of Cognitive Load Theory to Nursing Education. *International Journal of Nursing Education Scholarship*, 10(1), 1–9. Retrieved from [http://www.degruyter.com/dg/viewjournalissue/j\\$002fijnes.2013.10.issue-1\\$002fissue-files\\$002fijnes.2013.10.issue-1.xml;jsessionid=1E30D043F2B8CD51012150428A89975F](http://www.degruyter.com/dg/viewjournalissue/j$002fijnes.2013.10.issue-1$002fissue-files$002fijnes.2013.10.issue-1.xml;jsessionid=1E30D043F2B8CD51012150428A89975F)

- Holmqvist, K., Holsanova, J., Barthelson, M., & Lundqvist, D. (2003). *The Mind's Eye*. Elsevier. doi:10.1016/B978-044451020-4/50035-9
- Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Weijer, J. van de. (2011). *Eye Tracking: A comprehensive guide to methods and measures*. OUP Oxford. Retrieved from <https://books.google.com/books?hl=en&lr=&id=5rIDPV1EoLUC&pgis=1>
- Holmqvist, K., & Wartenberg, C. (2005). *The role of local design factors for newspaper reading behaviour – an eye-tracking perspective*. *Lund University Cognitive Studies*. Retrieved from <http://www.lucs.lu.se/LUCS/127/LUCS.127.pdf>
- Hornbæk, K., & Frokjaer, E. (2003). Reading patterns and usability in visualizations of electronic documents. *ACM Transactions on Computer-Human Interaction*, 10(2), 119–149. doi:10.1145/772047.772050
- Iahad, N., Dafoulas, G. A., Kalaitzakis, E., & Macaulay, L. A. (2004). Evaluation of Online Assessment: The Role of Feedback in Learner-Centered e-Learning. In *Proceedings of the 37th Hawaii International Conference on System Sciences*. Retrieved from http://www.stemfest.niu.edu/assessment/manual/_docs/study.pdf
- Iz, H. B., & Fok, H. S. (2007). Use of Bloom's taxonomic complexity in online multiple choice tests in Geomatics education. *Survey Review*, 39(305), 226–237. doi:10.1179/003962607X165195
- Jetton, T. L., & Alexander, P. A. (2001). *Learning from text: A multidimensional and developmental perspective / PEBC*. Retrieved from <http://www.pebc.org/research-and-practice/insights-from-research/i-content-area/learning-from-text-a-multidimensional-and-developmental-perspective/>
- Johnson, C. G., & Fuller, U. (2006). Is Bloom's taxonomy appropriate for computer science? In *Proceedings of the 6th Baltic Sea conference on Computing education research Koli Calling 2006 - Baltic Sea '06* (p. 120). New York, New York, USA: ACM Press. doi:10.1145/1315803.1315825
- Kuechler, W. L., & Simkin, M. G. (2003). How Well Do Multiple Choice Tests Evaluate Student Understanding in Computer Programming Classes? *Journal of Information Systems Education*, 14(4). Retrieved from [http://jise.org/Volume14/14-4/Pdf/14\(4\)-389.pdf](http://jise.org/Volume14/14-4/Pdf/14(4)-389.pdf)
- Lawless, K. A., Brown, S. W., & Mills, R. (2003). Knowledge, Interest, Recall and Navigation: A Look at Hypertext Processing. *Journal of Literacy Research*, 35, 911–934. doi:10.1207/s15548430jlr3503_5
- Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer*

- Interaction*, 7(1), 57–78. doi:10.1080/10447319509526110
- Liu, Z. (2005). Reading behavior in the digital environment: Changes in reading behavior over the past ten years. *Journal of Documentation*, 61(6), 700–712. doi:10.1108/00220410510632040
- Mayer, R. E., Dow, G. T., & Mayer, S. (n.d.). Multimedia Learning in an Interactive Self-Explaining Environment: What Works in the Design of Agent-Based Microworlds?
- Meiselwitz, G., & Sadera, W. A. (2008). Investigating the Connection between Usability and Learning Outcomes in Online Learning Environments. *MERLOT Journal of Online Learning and Teaching*, 4(2), 234–242.
- Moreno, R., & Mayer, R. (2007). Interactive Multimodal Learning Environments. *Educational Psychology Review*, 19(3), 309–326. doi:10.1007/s10648-007-9047-2
- NICCS. (2016). Curriculum Resources - Teaching Tools for Educators. *National Initiative for Cybersecurity Careers and Studies*. Retrieved from <http://niccs.us-cert.gov/education/curriculum-resources>
- Niederhauser, D. (2008). Educational hypertext research. In *Handbook of research on educational communications and technology* (3rd ed., pp. 192–210). New York: Lawrence Erlbaum Associates. Retrieved from [http://faculty.ksu.edu.sa/Alhassan/Hand book on research in educational communication/ER5849x_C016.fm.pdf](http://faculty.ksu.edu.sa/Alhassan/Hand%20book%20on%20research%20in%20educational%20communication/ER5849x_C016.fm.pdf)
- O'Brien, H. L., & Toms, E. G. (2008). What is user engagement? A conceptual framework for defining user engagement with technology. *Journal of the American Society for Information Science and Technology*, 59(6), 938–955. doi:10.1002/asi.20801
- O'Brien, H. L., & Toms, E. G. (2010). The development and evaluation of a survey to measure user engagement. *Journal of the American Society for Information Science and Technology*, 61(1), 50–69. doi:10.1002/asi.21229
- Paas, F. G. W. C. V. M. J. J. G. (n.d.). Instructional control of cognitive load in the training of complex cognitive tasks. . *Educational Psychology Review*. Dec94, 6(4).
- Parlangeli, O., Mengoni, G., & Guidi, S. (2011). The effect of system usability and multitasking activities in distance learning. In *Proceedings of the 9th ACM SIGCHI Italian Chapter International Conference on Computer-Human Interaction Facing Complexity - CHIItaly* (p. 59). New York, New York, USA: ACM Press. doi:10.1145/2037296.2037314
- Protopsaltis, A., & Bouk, V. (2005). Towards a Hypertext Reading/Comprehension Model. In *SIGDOC'05*.

- Quinn, C. N. (2005). *Engaging Learning: Designing e-Learning Simulation Games*. John Wiley & Sons. Retrieved from <https://books.google.com/books?hl=en&lr=&id=bJDzSsy3nCcC&pgis=1>
- Raina, S., Taylor, B., & Kaza, S. (2015). Security Injections 2.0: Increasing Engagement and Faculty Adoption using Enhanced Secure Coding Modules for Lower-level Programming Courses. In *9th World Conference on Information Security Education*. Hamburg, Germany: Springer International Publishing. doi:10.1007/978-3-319-18500-2
- Ramesh, A., Goldwasser, D., Huang, B., Daume, H., & Getoor, L. (2014). Uncovering hidden engagement patterns for predicting learner performance in MOOCs. In *Proceedings of the first ACM conference on Learning @ scale conference - L@S '14* (pp. 157–158). New York, New York, USA: ACM Press. doi:10.1145/2556325.2567857
- Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3), 372–422. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/9849112>
- Reeves, T. C. (2000). Alternative Assessment Approaches for Online Learning Environments in Higher Education. *Journal of Educational Computing Research*, 23(1), 101–111. Retrieved from <http://baywood.metapress.com/app/home/contribution.asp?referrer=parent&backto=issue,7,7;journal,107,195;linkingpublicationresults,1:300321,1>
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137–172. doi:10.1076/csed.13.2.137.14200
- Robinson, J. W. . J., & Crittenden, W. B. (1971). Learning Modules: A Concept for Extension Educators. *Journal of Extension*. Retrieved from <http://eric.ed.gov/?id=EJ070030>
- Rodriguez, B. C. P., & Armellini, A. (2013). Student engagement with a content-based learning design. *Research in Learning Technology*, 21. doi:10.3402/rlt.v21i0.22106
- Rowe, D. C., Lunt, B. M., & Ekstrom, J. J. (2011). The role of cyber-security in information technology education. In *Proceedings of the 2011 conference on Information technology education - SIGITE '11* (p. 113). New York, New York, USA: ACM Press. doi:10.1145/2047594.2047628
- Rudestam, K. E., & Schoenholtz-Read, J. (2010). *Handbook of Online Learning*. (K. E. Rudestam & J. Schoenholtz-Read, Eds.). SAGE Publications.
- Scouller, K. (2006). The influence of assessment method on students ' learning

- approaches : Multiple choice question examination versus assignment essay, 453–472.
- Shapiro, A. (1998). Promoting Active Learning: The Role of System Structure in Learning From Hypertext. *Human-Computer Interaction*, 13(1), 1–35. doi:10.1207/s15327051hci1301_1
- Sharmin, S., Špakov, O., & Rähä, K.-J. (2012, June 7). The Effect of Different Text Presentation Formats on Eye Movement Metrics in Reading. *Journal of Eye Movement Research*. doi:10.16910/jemr.5.3.3
- Shneiderman, B., Plaisant, C., Cohen, M., & Jacobs, S. (2010). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. (2010th ed.).
- Simola, J., Salojärvi, J., & Kojo, I. (2008). Using hidden Markov model to uncover processing states from eye movements in information search tasks. *Cognitive Systems Research*, 9(4), 237–251. doi:10.1016/j.cogsys.2008.01.002
- Southerland, J., & Nathaniel. (2010). Engagement of adult undergraduates: insights from the National Survey of Student Engagement. Retrieved from <http://content.lib.utah.edu/cdm/ref/collection/etd2/id/1290>
- Ssemugabi, S., & de Villiers, R. (2007). A comparative study of two usability evaluation methods using a web-based e-learning application. In *Proceedings of the 2007 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries - SAICSIT '07* (pp. 132–142). New York, New York, USA: ACM Press. doi:10.1145/1292491.1292507
- Stuart, I. (2004). THE IMPACT OF IMMEDIATE FEEDBACK ON STUDENT PERFORMANCE: AN EXPLORATORY STUDY IN SINGAPORE. *Global Perspectives on Account Education*, 1, 1–15. Retrieved from <http://feedback2.org/t/the-impact-of-immediate-feedback-on-student-performance-an-w97/>
- Tang, S., Reilly, R. G., & Vorstius, C. (2012). EyeMap: a software system for visualizing and analyzing eye movement data in reading. *Behavior Research Methods*, 44(2), 420–38. doi:10.3758/s13428-011-0156-y
- Taylor, B., & Azadegan, S. (2007). Teaching Security through Active Learning. In *Proceedings of Frontiers in Education: Computer Science and Engineering, Las Vegas* (pp. 1–6).
- Taylor, B., & Kaza, S. (2011a). Security injections: modules to help students remember, understand, and apply secure coding techniques. In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education* (pp. 3–7). ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=1999752>

- Taylor, B., & Kaza, S. (2011b). Security injections: modules to help students remember, understand, and apply secure coding techniques. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education - ITiCSE '11* (p. 3). New York, New York, USA: ACM Press. doi:10.1145/1999747.1999752
- Taylor, J. S. (2000). Using the World Wide Web in Undergraduate Geographic Education: Potentials and Pitfalls. *Journal of Geography*, 99(1), 11–22. Retrieved from <http://www.editlib.org/p/91942/>
- Teoh, B. S.-P., & Neo, T.-K. (2007). Interactive Multimedia Learning: Students' Attitudes and Learning Impact in an Animation Course. *Online Submission*, 6(4). Retrieved from <http://eric.ed.gov/?id=ED499660>
- Thalheimer, W. (2008). *Providing Learners with Feedback—Part 1: Research-based recommendations for training, education, and e-learning*. Somerville, Massachusetts, USA. Retrieved from http://willthalheimer.typepad.com/files/providing_learners_with_feedback_part1_may2008.pdf
- Thompson, E., Luxton-Reilly, A., Whalley, J. L., Hu, M., & Robbins, P. (2008). Bloom's taxonomy for CS assessment, 155–161. Retrieved from <http://dl.acm.org/citation.cfm?id=1379249.1379265>
- Tseng, M. (2008). The Difficulties That EFL Learners Have with Reading Text on the Web. *The Internet TESLJournal*, 14(2). Retrieved from <http://iteslj.org/Articles/Tseng-TextOnTheWeb.html>
- Turner, C. F., Taylor, B., & Kaza, S. (2011). Security in Computer Literacy- A Model for Design, Dissemination, and Assessment. In *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11* (p. 15). New York, New York, USA: ACM Press. doi:10.1145/1953163.1953174
- van der Kleij, F. M., Eggen, T. J. H. M., Timmers, C. F., & Veldkamp, B. P. (2012). Effects of feedback in a computer-based assessment for learning. *Computers & Education*, 58(1), 263–272. doi:10.1016/j.compedu.2011.07.020
- Wang, P.-Y., Vaughn, B. K., & Liu, M. (2011). The impact of animation interactivity on novices' learning of introductory statistics. *Computers & Education*, 56(1), 300–311. doi:10.1016/j.compedu.2010.07.011
- Wang, T. H. (2007). What strategies are effective for formative assessment in an e-learning environment? *Journal of Computer Assisted Learning*, 23(3), 171–186. doi:10.1111/j.1365-2729.2006.00211.x
- Wu, J.-H., Tennyson, R. D., & Hsia, T.-L. (2010). A study of student satisfaction in a

blended e-learning system environment. *Computers & Education*, 55(1), 155–164.
doi:10.1016/j.compedu.2009.12.012

Zaharias, P., & Poylymenakou, A. (2009). Developing a Usability Evaluation Method for e-Learning Applications: Beyond Functional Usability. *International Journal of Human-Computer Interaction*, 25(1), 75–98. doi:10.1080/10447310802546716

Zhang, D. (2010). Interactive Multimedia-Based E-Learning: A Study of Effectiveness. Retrieved from
http://www.tandfonline.com/doi/abs/10.1207/s15389286ajde1903_3#.VLQzg3sbuoh

Curriculum Vitae

Sagar Raina

Website: <http://triton.towson.edu/~csraina>

SPECIFIC Research interests

Cybersecurity Education
Human Computer Interactions
Computer Science Education
Educational Psychology
Learning Sciences

specific teaching interests

Introductory Programming
Human Computer Interactions
Database Management Systems
Web Development

Education

D.Sc. in Information Technology (expected May 2016)

Jess & Mildred Fisher College of Science and Mathematics, Towson University, Towson, MD

Dissertation topic: Examining Student Learning and Engagement Using Segmented and Interactive Modules: A Case Study in Cybersecurity Education
Dissertation Committee: Dr. Siddharth Kaza, Dr. Blair Taylor, Dr. Jinjuan Feng, Dr. Gabriele Meiselwitz
Courses: Data Structures & Algorithms, Database Management Systems, Operating Systems, Human Computer Interactions (HCI), Foundations of Instructional Technology, Statistics Research Design and Analysis
Overall GPA: 3.76/4.00

M.S. in Applied Information Technology

Towson University, Towson, MD, 2010-11

Concentration: Database Management Systems
Courses: Advanced Database Management Systems, Business Data Communications, Client/Server-Side Programming Web, Networks Architecture & Protocols, Information Technology Project Management, Information Technology Infrastructure, Information Technology & Business Strategy, Systems Development Process
Overall GPA: 3.81/4.00

Bachelor of Engineering, Information Technology

University of Mumbai, Bombay, India 2003-07

RESEARCH and PROFESSIONAL EXPERIENCE

Research Assistant (funded by NSF)

Department of Computer & Information Sciences, Towson University, 2012-present

Funded Projects:

Security Injections @Towson; Security Ambassadors Program; and Cyber4All

PI – Siddharth Kaza, Blair Taylor

Assist principal investigators in conducting literature review, research design, data collection and analysis.

Assist investigators in writing grants.

Design and develop segmented and interactive security injection modules using Django web-framework.

Setup and maintain linux based web-server to host 40 security injections modules with high traffic requests for django and wordpress web-frameworks.

Lead the group of graduate and undergraduate students and assess their weekly tasks.

Junior Software Developer

Social Solutions Inc. Baltimore, MD, 2011-12

Designed and developed databases, procedures, and triggers using Microsoft SQL server 2008.

Developed crystal reports using SAP based Business Intelligence (BI) tools.

Developed and maintained ASP.NET based application “Efforts to Outcome” (ETO).

Graduate Assistant

School of Emerging Technologies and department of CIS, Towson University, Towson, 2010-11

Set up and maintained computer security laboratory.

Broadcast and monitored online lectures using the mediasite software system.

Graduate Assistant

Department of Computer & Information Sciences, Towson University, Towson, 2010-11

Assisted investigators of Security Injections project in preparing secure coding modules in Java, C++, and Python.

Designed, developed and maintained Security Injections website using wordpress.

Policy Research Trainee

Department of Information Technology, Government of Himachal Pradesh, India, 2008-09

Analyzed several departments of Himachal Pradesh government for their Business Process Automation (BPA).

Inspected Information Technology Enabled Services (ITES) at various E-Governance centers in Himachal Pradesh.

Conducted research on Information Technology in government schools of Himachal

Pradesh.

Teaching Experience

Instructor – *Information and Technology for Business (COSC 111)*

Department of Computer & Information Sciences, Towson University, Fall 2012 –Spring 2015

Semester	Teaching Evaluations	Class Size
Fall2012	4.13/5.00	30
Spring 2013	4.96/5.00	27
Fall 2013	4.21/5.00	29
Spring 2014	4.69/5.00	29
Fall 2014	4.31/5.00	30
Spring 2015	4.88/5.00	20

Instructor – *General Computer Science (COSC 175)*

Department of Computer & Information Sciences, Towson University, Fall 2015
(Class Size – 28)

Semester	Teaching Evaluations	Class Size
Fall2015	4.17/5.00	27

Instructor – *Computer Science I (COSC 236)*

Department of Computer & Information Sciences, Towson University, Spring 2016
(Class Size – 30)

Conference publications

Full length papers

Raina, S., Bernard, L., Taylor, B., & Kaza, S. Using Eye-tracking to Investigate Content Skipping: A Study on Learning Modules in Cybersecurity. In IEEE International Conference on Intelligence and Security Informatics(2016). Tucson, Arizona (accepted).

Raina, S., Kaza, S., and Taylor, B. Security Injections 2.0: Increasing Ability to Apply Secure Coding Knowledge using Segmented and Interactive Modules in CS0. SIGCSE, (2016), Memphis, Tennessee, USA .

Raina, S., Taylor, B., and Kaza, S. Security Injections 2.0: Increasing Engagement and Faculty Adoption using Enhanced Secure Coding Modules for Lower-level Programming Courses Proceedings of the Ninth World Conference on Information Security Education, (2015), Hamburg, Germany.

Raina, S., Kaza, S., and Taylor, B. Segmented and Interactive Modules for Teaching Secure Coding: A Pilot Study. International Conference on e-Learning e-Education and Online Training, (2014), Washington DC, USA.

Posters

Raina, S., Taylor, B., and Kaza, S. Cyber4All: Increasing Cybersecurity Awareness,

Knowledge Retention and Engagement using Enhanced Security Injection Modules in Computer Literacy Courses. 19th Colloquium for Information Systems Security Education (CISSE 2015), Las Vegas, USA.

Taylor, B., Dudley, A., Santoro, M., and **Raina, S.** Secure Programming Logic Aimed at Students in High School. 19th Colloquium for Information Systems Security Education (CISSE 2015), Las Vegas, USA.

Raina, S., Taylor, B., and Kaza, S. Security Injections 2.0: Using Segmentation, Instant-feedback, and Auto-grading to Enhance Secure Coding Modules for Lower-level Programming Courses. SIGCSE, (2015), Kansas City, Missouri, USA.

Raina, S., Taylor, B., and Kaza, S. Interactive E-Learning Modules for Teaching Secure Coding: A Pilot Study. SIGCSE, (2014), Atlanta, Georgia, USA.

PRESENTATIONS & TALKS

Enhanced Security Injection Modules for Teaching Cybersecurity. 30th Anniversary (2014), Department of Computer and Information Sciences, Towson University, MD, USA

Building Security In: Injecting Security throughout the Undergraduate Computing Curriculum. Graduate Expo (2011), Towson University, MD, USA

PROFESSIONAL ACTIVITIES

Conference reviewer - International Conference on e-Learning e-Education and Online Training, (2014), Washington DC, USA

Student volunteer – SIGCSE 2014, 2015; IEEE Intelligence and Security Informatics (ISI) 2015

AWARDS AND HONORS

- **Graduate Student Achievement Award** – Graduate Students Association, Towson University (2015)
- **Vice President**, Doctoral Students Computer Science Association, Towson University (Fall 2015 - Present)
- **Graduate Student Travel Award** – Computer & Information Sciences, Towson University (2015)
- **Graduate Student Travel Award** – Graduate Students Association, Towson University (2015)
- **Graduate Student Travel Award** – Graduate Students Association, Towson University (2014)
- **Graduate Student Travel Award** – Computer & Information Sciences, Towson University (2014)
- **Graduate Student Travel Award** – The George Washington University (2013)
- **Doctoral Teaching Assistantship**, Computer & Information Sciences (Fall 2012 – Present)
- **Doctoral Research Assistantship**, Computer & Information Sciences (Fall 2012 - Present)
- **Graduate Assistantship**, Computer & Information Sciences, Towson University (Fall 2010 – Summer 2011)
- **Graduate Assistantship**, Center for Applied Information Technology, Towson University (Summer 2010 – Summer 2011)

TECHNICAL SKILLS

- **Programming Languages:** Java, C/C++, Python, C#
- **Web Technologies:** HTML, CSS, Javascript, Django, ASP.NET
- **Database Technologies:** MySQL, MS SQL server, Oracle, Sybase, SQL, PL/SQL

-
- **Operating Systems:** Linux – Ubuntu, Centos; Windows – XP, 7, 8
 - **IDE:** Netbeans, eclipse, visual studio
 - **Statistical Tool:** SPSS

