

This presentation was given at the 2023 National Cybersecurity Education Colloquium

Eye Tracking Technologies to Analyze and Visualize the Behavior of Secure Coders

Daniel Davis

Doctor of Philosophy - Computer Science - UAH - 2023

Research Advisor / Committee Chair - Dr. Feng Zhu



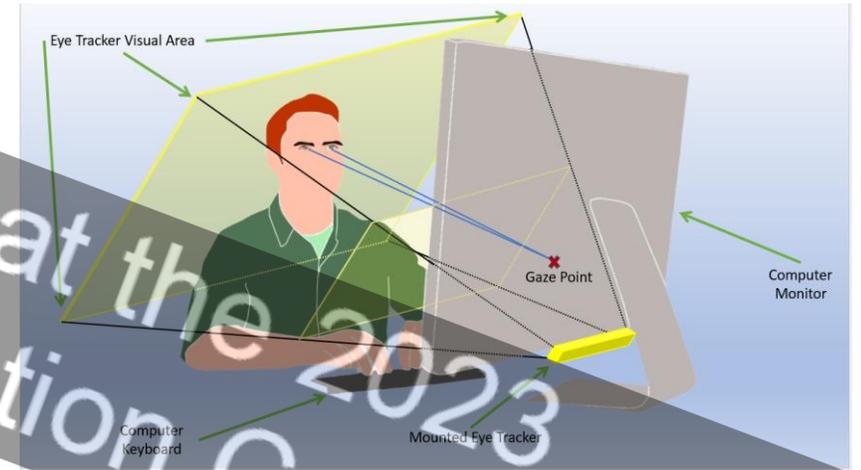
THE UNIVERSITY OF
ALABAMA IN HUNTSVILLE

Introduction

- Secure coders techniques and procedures
- Costly coding security flaws in applications
- Common Weakness Enumeration (CWE) repository
- Build hands-on secure coding learning modules
- Eye tracking technologies to capture behaviors
- Literature survey of existing visualization analysis
- Static vs (linear and non-linear) dynamic stimuli
- Eye tracking design and analysis framework
- Decision matrix for designing experiment
- Limitations of current visualization methods used in eye tracking
- Scrolling stimuli
- Participant-editable stimuli
- Creating Areas of Interests (AOIs)
 - How do coders find and mitigate security flaws
 - How developers read/write code, utilize security tools, and read instructions
- Transitions among eye tracking stimuli and between software application

Research Contributions

- Classification of the goals, objectives, participant tasks, and visualization techniques in distinct stages of the SDLC for eye tracking
- Understanding of secure coders' behaviors with multiple types of visualizations of distinct aspects in secure coding and over a timeline
- At the low level, we process eye movements, the speed of movement, the duration of eye fixation, and changes in pupil sizes
- At the medium level, we examine participants' eye gaze at the application and source code files or function level
- At the high level, we present participants' secure coding patterns and strategies
- We propose swimlane diagrams, state transition diagrams, and pupil size fluctuation diagrams
- Developed our Eye Tracking Design and Analysis Framework for software development with a focus on secure coding
- A decision matrix for mapping objectives/tasks in the SDLC to specific aspects of eye tracking design, analysis, and comparison
- Guide on the type of software tasks and eye tracking stimuli to present to participants



Publications

- The Study of Cryptographic Algorithms and Performance Measurements Across Heterogeneous Devices
 - Computer Science and Education in Computer Science (CSECS)
 - July 2016
 - Pages 205 - 219
- Understanding and Improving Secure Coding Behavior with Eye Tracking Methodologies
 - Association for Computing Machinery Southeast (ACMSE)
 - April 2020
 - Pages 107 - 114
- Analysis of Software Developers' Coding Behavior: A Survey of Visualization Analysis Techniques Using Eye Trackers
 - Computers in Human Behavior Reports
 - August 2022
 - Pages 1 - 28
- Eye Tracking Technologies to Visualize Secure Coding Behavior
 - Array
 - September 2022
 - Pages 1 - 34

Understanding and Improving Secure Coding Behavior with Eye Tracking Methodologies

Daniel Davis
University of Alabama in Huntsville
Computer Science Department
Huntsville, AL, USA
dkd0004@uah.edu

Feng Zhu
University of Alabama in Huntsville
Computer Science Department
Huntsville, AL, USA
fz0001@uah.edu

ABSTRACT

Secure coding is a mission that cannot be neglected as

KEYWORDS

Cyber Threats, Education, Coding Behavior, Software Vulnerabilities, Cyber-Attacks, Security

Computers in Human Behavior Reports 7 (2022) 100213

Contents lists available at ScienceDirect

Computers in Human Behavior Reports

ELSEVIER journal homepage: www.elsevier.com/locate/computer-human-behavior-reports

Analysis of software developers' coding behavior: A survey of visualization analysis techniques using eye trackers

Daniel Kyle Davis^a, Feng Zhu^b

^aUniversity of Alabama in Huntsville, Computer Science Department, 301 Sparkman Dr NW, Huntsville, AL, 35899, USA

ARTICLE INFO

Keywords:

Software development
Software coding behavior
Hands-on coding exercises
Eye tracking
Eye tracking user interactive stimuli
Eye tracking visualizations survey

ABSTRACT

The coding behavior of a software developer is not easily gathered and investigated. One accurate and objective approach is to utilize eye tracking techniques to capture how developers write code, use tools, and read natural language documents and instructions. Beneficial insight is to visualize the eye gases that capture coding behavior. A software developer tasks, and individual actions, the complexity to design eye tracking experiments and to analyze the eye gases. Our systematic literature survey, focuses on published methods for multiple types of static and dynamic eye tracking stimuli, especially techniques that use multiple participant-editable types of stimuli presented once to allow for a realistic coding experience. We propose an eye tracking design and analysis framework, which breaks down the various stages of software coding that require using different programming skills. Our decision matrix maps objectives for software programming to analysis techniques for comparing the eye gases among software developers. Finally, we discuss the limitations of current visualization methods, specifically for user controlled dynamic stimuli consisting of writing source code stimuli. We propose additional visualization techniques to help researchers investigate behavior of software developers' while they work software coding tasks.

1. Introduction

Software development involves the creation of software applications through many activities including requirements gathering, designing modules, coding, testing, deploying and finally maintaining a software application. The computing instructions contained in a software application are typically coded by a software developer to meet the requirements and design of the software application while supporting long-term maintainability of the software. Software developers improve their ability and skills to develop quality software application with more experience as they learn patterns and techniques of writing and debugging software throughout their career. Software developers may include algorithm coders, software testers, and software security developers. A developer may work in a single stage or multiple stages of the Software Development Lifecycle (SDLC) (Langer, 2012). However, the patterns and techniques that a software developer utilizes when searching, writing, and testing software code are not easily collected by researchers in an objective and non-intrusive manner. Techniques that can objectively collect and measure software coders' behavior while

remaining unintrusive can allow a researcher to observe and discover where and when software developers' focus on key components in the SDLC such as in the phase of software coding and software testing for coding flaws.

Questionnaires, verbal interview reviews, think-aloud sessions, behavioral retrospectives and other self-reporting methods are helpful, but they cannot provide the objective and detailed information that is needed to study the techniques and approaches developers utilize in software development activities (Lai et al., 2013). Developers may rapidly switch between different files and tools, may have multiple types of stimuli content on a computer display at the same time, and may modify their code. Typical tasks include the reading of multiple source code files, using web browsers, code editing tools, and static code analysis tools. The analysis of their behavior may require examining the stimuli at the programming statement line level, function level, the file level, and the application level. The details cannot simply be portrayed in questionnaires, retrospectives, or think-aloud sessions. Eye tracking technologies allow researchers to record and investigate the eye gazes and patterns of participants, and therefore observe the

Abbreviation: Areas of Interest (AOIs), Integrated Development Environment (IDE), Simple Moving Average (SMA), Software Development Lifecycle (SDLC).
^a Corresponding author.
E-mail addresses: dkd0004@uah.edu (D.K. Davis), fz0001@uah.edu (F. Zhu).

<https://doi.org/10.1016/j.chbr.2022.100213>

Received 2 January 2022; Received in revised form 16 May 2022; Accepted 12 June 2022

Available online 16 June 2022

2451-9588/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

THE STUDY OF CRYPTOGRAPHIC ALGORITHMS AND PERFORMANCE MEASUREMENTS ACROSS HETEROGENEOUS DEVICES

Daniel Davis, Feng Zhu

University of Alabama in Huntsville, Computer Science Department
in the College of Science

Abstract: Confidential information can be safely sent between various types of devices over insecure communication channels by using cryptography. Typically, students are taught how cryptography algorithms are designed and how they should be safely utilized.

Array 15 (2022) 100241

Contents lists available at ScienceDirect

Array

ELSEVIER journal homepage: www.elsevier.com/locate/array

Eye tracking technologies to visualize secure coding behavior

Daniel Kyle Davis^a, Feng Zhu^b

^aUniversity of Alabama in Huntsville, Computer Science Department, 301 Sparkman Dr NW, Huntsville, AL, 35899, United States

ARTICLE INFO

Keywords:
Cybersecurity secure coding
Hands-on education learning
Software development coding behavior
Eye tracking
Eye tracking user interactive stimuli
Software vulnerabilities

ABSTRACT

Secure coders' experiences and performances vary greatly and any missed security flaws in source code may lead to costly consequences. Their behavior to analyze source code and develop mitigation techniques is not well understood. Our objective is to gain insight into the strategies and techniques from both novice and experienced developers. Proper understanding can help us to inform inexperienced coders to efficiently and accurately approach, discover, and mitigate security flaws. Our research relies upon eye tracking hardware and software to collect and analyze the eye gases. Unlike existing approaches, we incorporate a wide range of tasks simultaneously reading documentation, writing code, and using security coding analysis tools. We analyze both static and dynamic (interactive) stimuli in a realistic software development environment. Our pictorial visualizations represent a coder's eye gases that visually demonstrates their behavior and patterns. In addition, we provide the full content of the stimuli that participants observed. This allows for investigating the behavior at a range of tasks for a single participant and between participants. Our secure coding tasks include reading documentation, reading source code, and writing source code for a web application as well as utilizing security code scanning tools. Our tasks include: (1) novel visualization techniques to present transitions among components within and between applications, and (2) presentation of coders' attention levels during secure coding by investigating the change of pupil sizes. The eye tracking collection and analysis techniques support both modifiable stimuli and stimuli presented in different sequences based upon individual participant's behavior.

1. Introduction

Secure coders often examine various aspects of a programing code, syntax, semantics, program flow, and overall program execution. They typically focus on finding security weaknesses and then determine the appropriate mitigation techniques. With diverse skills, secure coders may approach discovering and fixing vulnerabilities very differently and inefficiently. While secure coding becomes increasingly important, secure coding behavior is not well understood. Understanding secure coding behavior and where challenges occur during coding can help us understand where to improve learning content and therefore improve secure coders abilities.

Existing methods of investigating behaviors often utilize think-aloud sessions, questionnaires, or verbal reviews, but secure coders verbalized is stored in skill memories, which cannot always be verbalized [1]. Therefore, it is difficult to collect and analyze behavior and patterns via questionnaires or verbal reviews in a non-bias or objective manner. Direct observations do not require participants to recall or verbally explain their techniques during secure coding, but it is unnatural and

distracting to participants. [1]. A discreet and unobtrusive technique is needed to observe secure coders behaviors while not requiring participants to recall or verbally explain their techniques during secure coding.

Eye tracking technologies allow researchers to observe participants' eye gazes objectively. The analysis of eye tracking results allows for investigating participants' reading patterns and behavior across distinct types of stimuli without requiring participants to verbally communicate their actions. Most existing eye tracking technologies focus on top-down and left-to-right reading pattern but reading source code and performing secure coding activities does not follow the standard text reading patterns or even general software coding patterns. Existing visualization methods for eye gaze analysis have limitations for eye tracking stimuli that is user driven and when multiple types of eye tracking stimuli are presented at a same time.

The fundamental challenge stems from user-controlled eye tracking stimuli. Secure coders may read paragraphs of documentation and guidelines; they may read and write source code; and they may utilize security scanning tools. Often secure coders perform several of these tasks concurrently with an application window adjacent to other applications and rapidly switch focus between applications. Furthermore, the

^a Corresponding author.
E-mail addresses: dkd0004@uah.edu (D.K. Davis), fz0001@uah.edu (F. Zhu).

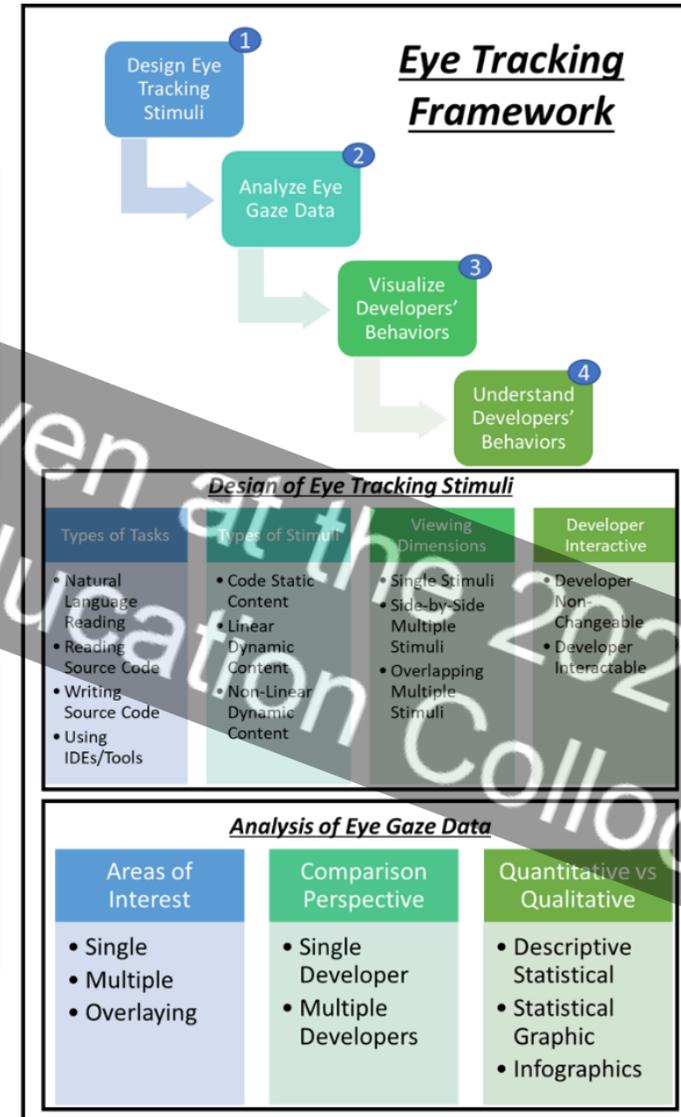
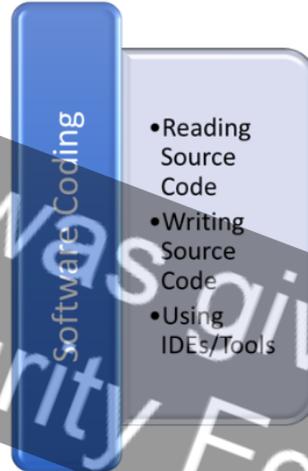
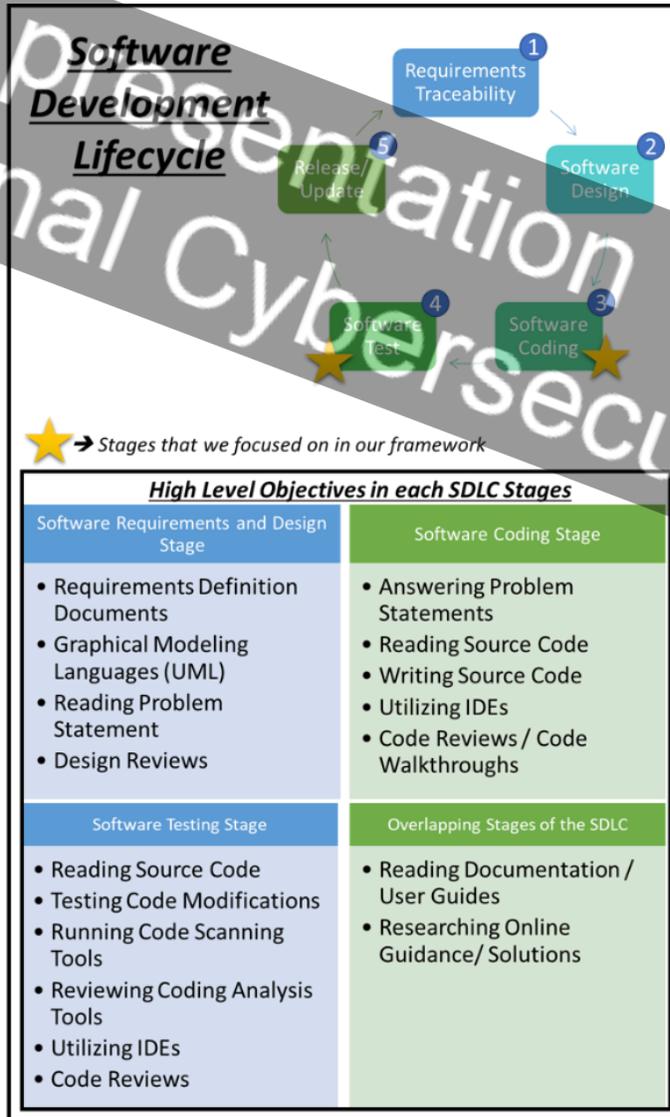
<https://doi.org/10.1016/j.array.2022.100241>

Received 5 April 2022; Received in revised form 27 July 2022; Accepted 28 July 2022

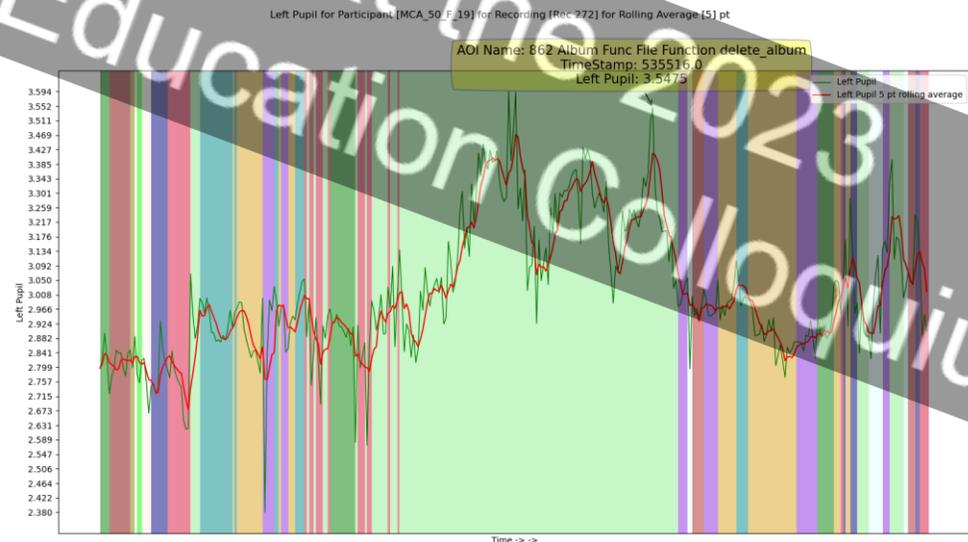
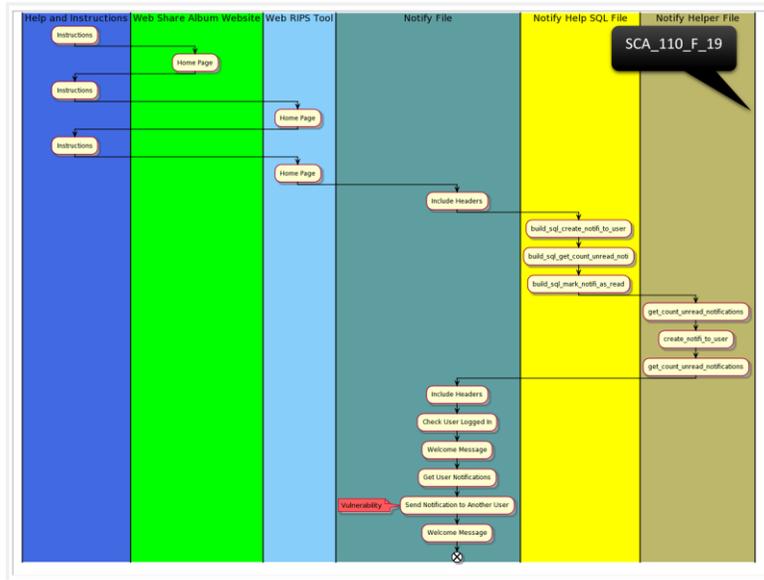
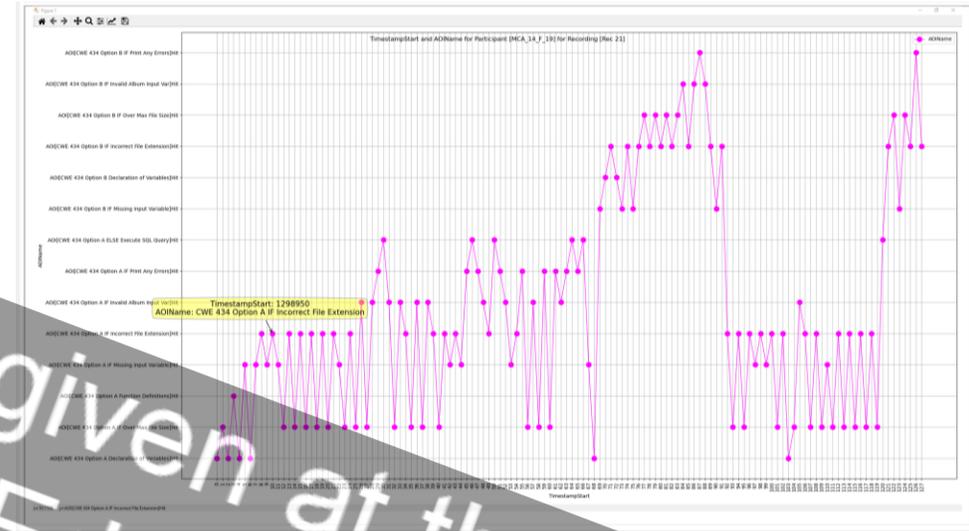
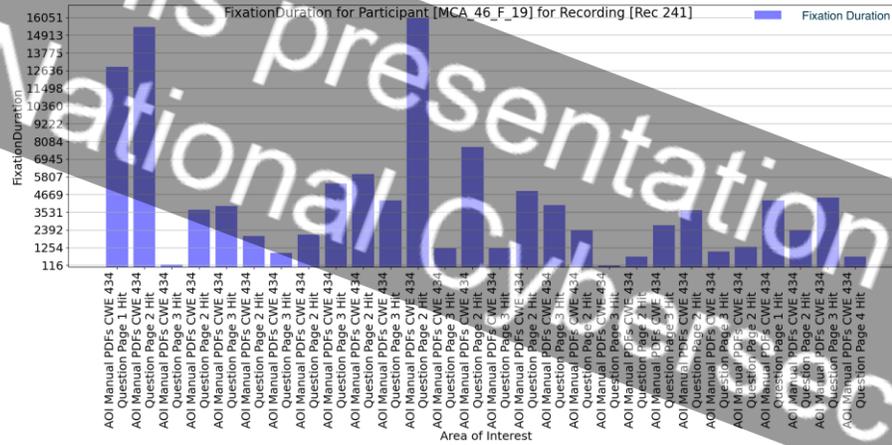
Available online 18 August 2022

2590-0056/© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

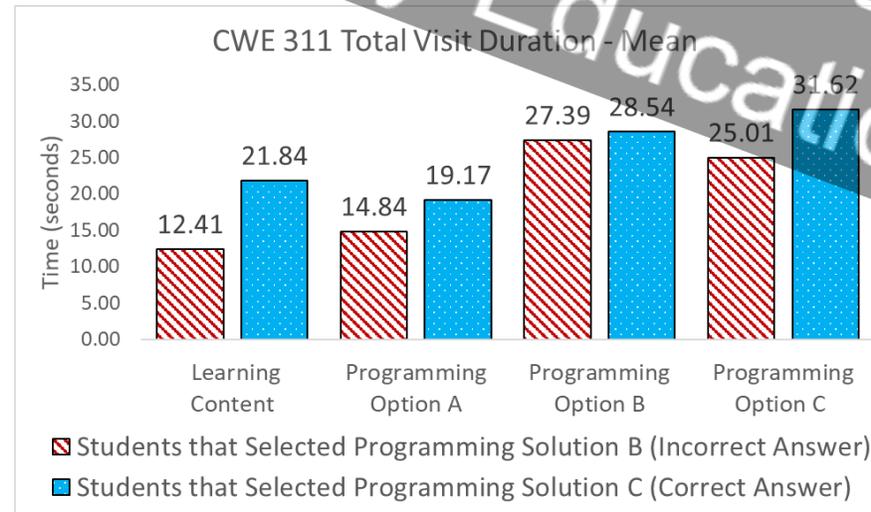
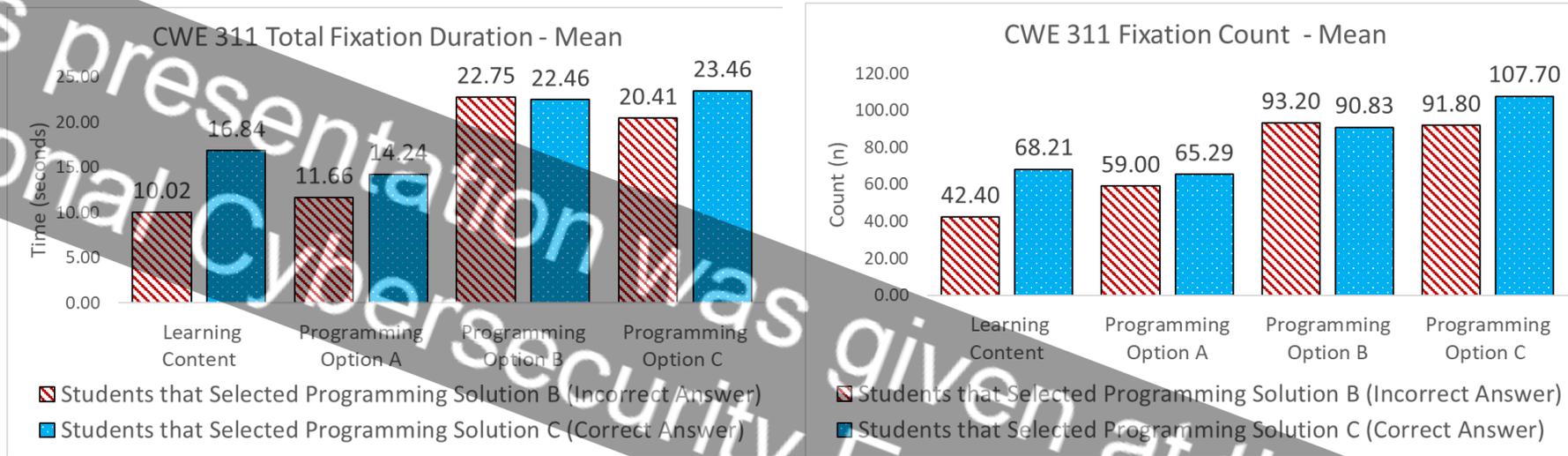
Study Design / Methods



Results



Secure Coder Responses CWE-311



Ongoing Work

- Our focus has been on analyzing the data at both the low-level and high-level
- One area we are investigating is to compare the behaviors of those that answered correctly with those that did not answer correctly
 - Methods being explored
 - Manually Analysis
 - Automated Analysis (low-level or high-level) (machine learning)
 - CWE Problems being explored
 - True/False if Software Flaw - CWE-443 or CWE-73
 - Programming Problem - CWE-862 or CWE-22
- Analyze reading patterns between novice and expert secure coders
 - Majority of our data is with novice secure coders

This presentation was given at the 2023
National Cybersecurity Education Colloquium



QUESTIONS / COMMENTS

