Spring 5-19-2017

Development of Peer Instruction Material for a Cybersecurity Curriculum

William Johnson
University of New Orleans, wejohnso@uno.edu

Development of Peer Instruction Material for a Cybersecurity Curriculum

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Computer Science
Information Assurance

by

William E. Johnson

B.A. Millsaps College, 2011

May, 2017

Dedication

This thesis is dedicated to my family, who have been supportive from the beginning.

Acknowledgment

I would like to thank Dr. Irfan Ahmed as my major professor and the rest of the Information Assurance faculty at the University of New Orleans for providing me with this opportunity as well as giving me a place within the concentration. I want to thank Dr. Golden G. Richard III for inspiring me to enter into the University of New Orleans to obtain a Master's degree for information security as well as guiding me through my time here. I also want to thank Dr. Vassil Roussev for support and guidance through my graduate coursework. They have all been invaluable throughout my time spent here.

I want to extend my thanks to Joe Sylve for the LaTeX template–it's been indispensable for writing this thesis.

I would also like to thank Dr. Cynthia B. Lee of Stanford–her collaboration and assistance with this project is immensely appreciated.

Finally, I would like to thank Allison Luzader for her collaboration.

This thesis is a continuation of work published by William Johnson, Allison Luzader, Dr. Irfan Ahmed, and Dr. Vassil Roussev of the University of New Orleans Department of Computer Science; Dr. Golden G. Richard III of Louisiana State University's Center for Computation and Technology; and Dr. Cynthia B. Lee of the Stanford computer science department [5].

The research supporting this thesis was funded by the National Science Foundation under award number 1500101.

Table of Contents

List of Figures

List of Tables

Abstract

Cybersecurity classes focus on building practical skills alongside the development of the open mindset that is essential to tackle the dynamic cybersecurity landscape. Unfortunately, traditional lecture-style teaching is insufficient for this task. Peer instruction is a non-traditional, active learning approach that has proven to be effective in computer science courses. The challenge in adopting peer instruction is the development of conceptual questions. This thesis presents a methodology for developing peer instruction questions for cybersecurity courses, consisting of four stages: concept identification, concept trigger, question presentation, and development. The thesis analyzes 279 questions developed over two years for three cybersecurity courses: introduction to computer security, network penetration testing, and introduction to computer forensics. Additionally, it discusses examples of peer instruction questions in terms of the methodology. Finally, it summarizes the usage of a workshop for testing a selection of peer instruction questions as well as gathering data outside of normal courses.

Keywords: Peer instruction, cybersecurity, computer security, digital forensics, reverse engineering, network penetration testing

Chapter 1

Introduction

Cybersecurity is one of most strategically important areas of computer science and also one of the most difficult disciplines to teach effectively. The escalating reliance on IT tools in all aspects of social life is leading to ever increasing costs in cybersecurity failures. The vast majority of these failures are the result of poor understanding of the security landscape, an overly abstract view of important computing concepts, and an inability to adapt to new threats.

Engineering a secure IT system, in addition to technical skills, requires out-of-the-box thinking that takes into account the incentives and capabilities of both the attacker and the defender. To be effective, a cybersecurity professional must be flexible and creative, able to quickly adapt within the fast-changing security landscape. In such a dynamic environment, education is a continuous process and requires the mindset that learning on the job is part of the daily routine. It is imperative that we find methodologies that can reliably improve learning outcomes and develop workforce proficiency in these strategically important areas.

Unfortunately, the traditional lecture is a poor match for the need to develop students into creative thinkers and lifelong learners, and this is especially true for cybersecurity educationboth within and outside of academia. This is the direct result of an over-emphasis on *specific* (lifespan limited) technical skills without attention to fundamental conceptual underpinnings. Other challenges include a lack of technical depth, and impatience towards developing broader analytical skills.

One of the main difficulties in delivering the necessary educational outcomes is that students need to experience a significant number of realistic situations before they can appreciate practical security problems and start to reason about the corresponding situations. In other words, presenting the underlying concepts is a necessary part of the job for the

instructor, but it is well short of sufficient. One of the biggest instructional challenges is to balance the requirements of discussing concepts and building hands-on skills within the confines of a semester. In our view, the only way to accomplish this is to encourage the students to do more preparation *before* coming to class (and to continue this preparation well after class has ended), and to actively participate in class discussions with their peers.

Motivating students to study before class has been a challenge in other disciplines and one of the more promising solutions that has emerged is the concept of *peer instruction*. This teaching paradigm was introduced by Eric Mazur, a physicist at Harvard University, who realized that his students could pass their traditional, formulaic problem-solving exam but have little conceptual understanding of Newtonian physics [2], [16]. When confronted with new types of questions on the same concepts, they were simply not able to adapt.

In a peer instruction classroom, lecture is interspersed with multiple-choice questions known as ConcepTests, which are designed to provoke deep conceptual thinking in students and engage them in meaningful discussion with their peers. Peer instruction has been shown to improve outcomes in several scientific disciplines, such as physics, computer science, and biology. In computer science (CS), it has shown promising results such as halving failure rates in four different courses [9] and increasing retention in the major [12].

Although reports from the field show the successes of peer instruction in CS, the current primary focus has been limited to theoretical and introductory programming courses [6]. In our experience, there are substantial differences between teaching a standard CS course and an advanced cybersecurity one. For instance, cybersecurity teaching is expected to transform students mindsets for increased adaptability and analytical skills for assessing dynamic risks and defense strategies. Unfortunately, peer instruction is not widely used in cybersecurity education and a major challenge in adapting peer instruction is the development of in-class, conceptual questions appropriate to the unique goals of cybersecurity education.

Over the past two years, we have made significant efforts and developed 279 peer instruction questions for three cybersecurity courses: introduction of computer security, net-

work penetration testing, and introduction to computer forensics. This thesis analyzes these questions and identifies a systematic methodology for developing peer instruction questions. Furthermore, it provides five examples of the peer instruction questions in cybersecurity.

In addition, we discuss our experience holding a condensed computer forensics workshop for experimentation of usage of peer instruction with material focusing on the FAT32 file system, file carving, and forensic artifacts of the Windows registry.

The rest of the discussion is organized as follows: Chapter 2 provides some background for peer instruction. Chapter 3 presents a methodology for developing peer instruction questions, and chapter 4 provides examples of peer instruction questions for four major cybersecurity areas. Chapter 5 provides a comprehensive analysis of the peer instruction questions recently developed for three cybersecurity courses. Chapter 6 details a small workshop we held primarily to test a selection of peer instruction questions and gauge student opinion. Chapter 7 discusses some prior work on peer instruction within computer science. Finally, chapter 8 concludes the discussion and details potential future work.

Chapter 2

Peer Instruction Background

2.1 Peer Instruction Methodology

The peer instruction method divides the lecture period into small presentations. Each presentation focuses on a central point and is typically followed by a series of the following activities:

- A conceptual question is asked to students, who are then given two to three minutes to formulate individual answers and report them to the instructor. Typically, the question is in multiple-choice format, enabling aggregation of student response data by the instructor.

- If a mix of correct and incorrect answers is received, students are further encouraged to discuss their answers with others sitting around them. The discussion may last three to four minutes. The goal of the discussion is to present the fundamental reasons behind the answers and for students to convince each other of the correctness of their own answers.

- Students are then asked to stop discussion and polls for their answers are performed again to observe how their opinions were influenced by the discussion in the previous step.

- After reviewing the poll results, the instructor decides to either move on to the next concept or present the correct solution with more explanation, as needed.

Peer instruction requires students to be better prepared for each class. The instructor provides reading material on the topic to be covered in the class and the students have to read

the material before the class, allowing them to better understand the presentations and respond to conceptual questions. The students are also given a quiz to solve after reading the material, and some incentives (such as bonus marks) are associated with each quiz. This approach encourages students to go through the material carefully and to be prepared.

2.2 Peer Instruction Outcomes in CS

More recently, peer instruction has been introduced in computer science, and research has shown that computer science students both value peer instruction and also recommend that more instructors use it at both small colleges and large schools [17], [15]. Research also shows that instructors who use a peer instruction approach in their classrooms find it quite effective [10]. Essentially, the real learning occurs during discussions between students when a conceptual question is asked to them [11].

Research also shows that students who have learned through peer instruction achieve 6% higher grades on their final exams than students in a lecture-centric standard teaching environments [18]. Peer instruction has shown effective results in reducing failure rates by 61% on average in four computer science courses (CS1, CS1.5, Theory of Computation, and Computer Architecture) [9]. It has also shown a 31% improvement in the retention of students in a computer science major [13].

Chapter 3

Question Development Methodology for Peer Instruction

3.1 Challenges for Developing Questions

In our experience, there are a number of challenges that arise when developing peer instruction questions. In particular, this section discusses two challenges that are encountered frequently.

Quiz vs. Peer Instruction Questions The main challenge is the development of multiple-choice questions to facilitate peer discussion without the questions seeming non-conceptual. Questions reserved for quizzes rather than peer instruction need be less conceptual, as they are typically used to test knowledge rather than the application of knowledge. It can be difficult to step away from that for conceptual questions without these questions seeming too simple.

Plausible Incorrect Answers for a Peer Instruction Question Similarly, there is a difficulty in creating incorrect answers for peer instruction questions that seem plausible. The trolling for misconceptions" and similar question tactics by Beatty et al. [1] assist in fulfilling this issue, though it is occasionally also difficult to realize potential misconceptions.

3.2 Overview of the Methodology

We have identified a basic methodology for creating peer instruction questions systematically. Figure 3.1 illustrates four stages of the methodology: concept identification, concept trigger, question presentation, and question development. To develop a peer in-

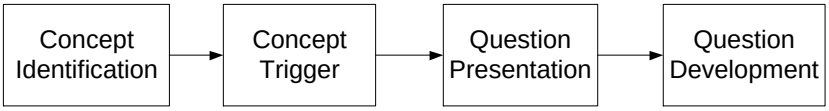


Figure 3.1: Overview of the methodology for developing peer instruction questions

struction question, the first stage is to identify a concept defining the main focus of a question, and then further (in the second stage) identify a concept trigger that is introduced in the question to provoke a students thinking process and set the desired direction of peer discussion. Deliberately introducing ambiguity in answer choices is an example of concept triggers (more examples are discussed in Table 6.1). Multiple concept triggers can be used for a question.

The third stage determines how the concept and the concept-trigger(s) (identified in last two stages) can be put together in a question for better presentation, and easier understanding. For instance, a question can be presented in a scenario, example, or diagram. The last stage of the methodology creates the question, including articulation and identification of multiple choices.

Chapter 4

Examples of Peer Instruction Questions

This chapter presents five examples of peer instruction questions representing four distinct cybersecurity areas: introductory cybersecurity concepts, computer forensics, reverse engineering, and network penetration testing. The first is typically taught in the traditional lecture format, whereas the latter three are intensive hands-on courses. Together, they form the basis of a broad skill set and perspective on security.

The chapter further discusses concept triggers and question presentation for each question. We borrow our concept triggers from Beatty et al. [1], and used them for the detailed analysis of our peer instruction questions (discussed in §5). Concept triggers (mostly used in our analysis) are briefly described in Table 6.1. Furthermore, we identify four question-presentation types after carefully analyzing our peer instruction questions: Examples, Scenario, Definitional, Feature, and Diagram.

Scenario questions present students with a situation, and require students to answer the question provided about the situation by examining the literal and implied details of that situation. *Example* questions simply provide describe a sample system or code–these are somewhat similar to scenario questions, but are more straightforward, as there is less interpretation required for students to understand and respond to the question. *Definitional* questions are even simpler–they deal strictly with the definition of a concept, and are used best when attempting to differentiate between two or more particularly similar concepts. *Diagram* questions present students with a diagram and ask them to make interpretations based on the visual–these must not simply be code snippets; they must have a strong visual component. Finally, *feature* questions deal with the components of a concept–they are questions that may ask whether a provided example has all of the required features of that concept, which feature is a major component of the concept, or which concept the provided features

best support. They are, essentially, similar but more specific than definitional questions, as they target

Concept triggers	Description
Compare and contrast	Compare multiple situations; draw conclusions from comparison
Interpret representations	Provide a situation that asks students to make inferences based upon the presented features
Identify a set or subset	Ask them to identify a set or subset fulfilling some criterion
Strategize only	Provide a problem; ask students to identify the best means of reaching a solution
Omit necessary information	Provide less information than is essential for answer; see if students realize this
Use "none of the above"	Provide an option to learn alternative understandings; use it occasionally as a correct answer
Qualitative questions	Questions are about the concepts and relationships rather than numbers or equations
Analysis and reasoning questions	Create questions that require significant decision-making, hence promote significant discussion
Trap unjustified assumptions	Answer choices are facilitated by potential unjustified assumptions made by the students
Deliberate ambiguity	Use deliberate ambiguity in questions to facilitate discussion
Trolling for misconceptions	Attempt to trap students with answers that require common misconceptions to choose

Table 4.1: Sample concept-triggers borrowed from Beatty et al. [1]. We use them for analyzing peer instruction questions

4.1 Introductory Cybersecurity Concepts

This example of a peer instruction question is used to introduce the key concepts of confidentiality, integrity, and availability (CIA triad). In class, the instructor initially presents some slides and explains the concepts. After giving a few examples, the instructor shows the peer instruction question. Students are given an opportunity to answer individually first. After all students have responded, students discuss their answers in small groups and come to a consensus. This approach gives students the opportunity to engage in discussion and problem solving with their peers to arrive at an answer.

One example of the peer instruction question could be: *An attacker deletes files on a system, denying access to system users. Which element of CIA triad is violated?* a) Confidentiality, b) Integrity, c) Availability, and d) None/Other/More than one of the above.

Concept Triggers: We can deconstruct this peer instruction question and identify some of the question design tactics by Beatty et al. that might be used to construct a question such as this one. First, by noting the question option D, this question introduces the usage of "none of the above" as well as "identify a set or subset", by allowing the students to select none of the above or any option that they may choose from the set.

Additionally, this question is qualitative rather than quantitative as it approaches the concept of the balance of the CIA (Confidentiality, Integrity, and Availability) triad and its components, representing the tactic of qualitative questions.

Finally, this question promotes the usage of "multiple defensible answers", as while it lends itself primarily to exhibiting a violation of availability (due to the phrase "denying access to system users"), this attack would secondarily be considered a violation of system integrity, and as such, choice C would be the primary answer, but choice B is also acceptable. Ultimately, the answer is D.

Question Presentation: This question is presented as an example. However, it can be elaborated in a scenario.

4.2 Digital Forensics

In *digital forensics*, the problem of comprehensive data recovery is a critical first step for most inquiries. In most cases, the formatting of a hard disk has a very small impact on the actual data content, because it simply overwrites file system metadata (making the data inaccessible via the OS interface). There is a common misconception, even among users with a strong technical background, that format operations permanently (and extensively)

destroy file content. In fact, with the exception of internal SSDs, most format operations destroy very little file content. One example question to make students think about it, could be: *Estimate the fraction of disk blocks affected by formatting the disk:* a) 100%, b) 65%, c) 20%, d) ¡5% (answer: d). Follow-up questions could lead them to the fact that the answer may be different, depending on the target medium–HDD vs. SSD vs. virtual disk.

The corresponding peer instruction lesson involves describing basic filesystem layouts, then asking students to take a position on what percentage of filesystem data would be destroyed by first formatting a USB flash drive using the FAT filesystem. The scenario is then expanded to include a sequence of other format operations using different filesystem types, including NTFS, ext3 on Linux, etc., against the same flash drive. Students are asked to agree how to visually depict how much data they predict is permanently destroyed. The instructor then uses a visual aid to illustrate exactly how much data is destroyed. The results are illustrated in Figure 4.1. Figure 4.1 illustrates data deletion (in red) as the flash drive is subjected to a number of format operations. Green areas remain untouched by format operations.

Concept Triggers: Deconstructing this question, it uses some interesting question tactics. First, formatting a disk can have vastly different effects depending on the filesystem type–this is a particularly important detail missing. Initially, this question can be noted as using "omit necessary information" as well as "trap unjustified assumptions", as any particular answer would need to assume a particular formatting scheme–trolling for misconceptions could work as well, if students assume that this operation is performed the same regardless of filesystem type. Finally, this question involves "deliberate ambiguity", and it is a "qualitative question". It is important to understand the power of this type of design tactic, as leading a discussion or even curriculum with a generalizing misconception such as this can be powerful when leading students to understand the difference amongst important specifics.

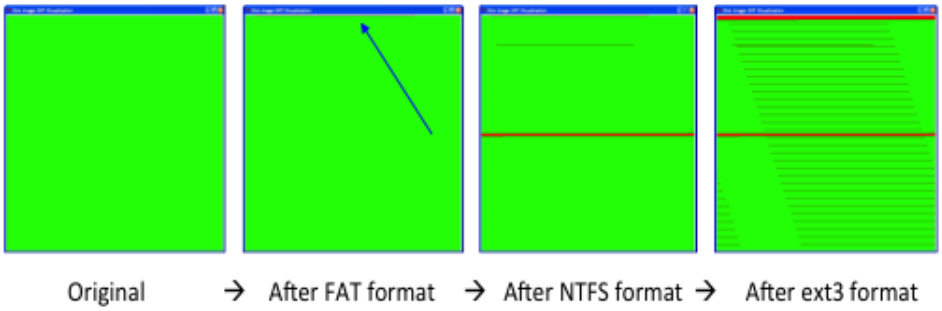


Figure 4.1: The results of formatting a flash drive with `FAT`, then `NTFS`, then the `EXT3` filesystem. Deleted regions are shown in red.

Question Presentation: This question could be identified primarily by *feature*, as it requires an understanding of disk formatting, and students will need to understand the features of that process to guess the correct percentage of blocks affected–even though this question intentionally leaves out a significant amount of information.

4.3 Reverse Engineering

In *reverse engineering*, it is important not only to understand assembly-level language, but it is also critical to know the specifics of its implementation on particular hardware. With assembly language, the programmer has fewer restrictions in terms of access to memory than in higher-level languages such as C or C++. In effect, this allows the programmer to easily create self-modifying code that overwrites other instructions or operands in memory. However, instruction prefetch caching introduces additional complexity. If code is modified in memory, the modification will persist, but if prefetch is enabled, this only applies to current control flow to an extent. In the case of a small number of instructions (16 bytes for the Intel 80486 prefetch queue), if an instruction modifies code that has previously been fetched, it will be executed as if it were not modified in memory. If prefetch caching is not enabled (as is the case when single-stepping in a debugger), modified code that falls in control flow will be executed as if it is modified, regardless of locality to the instruction pointer. An example question to explore this concept is, given a code example (shown in Listing 1) that presents code modification within prefetch range of control flow: *After executing these instructions*

```
1
2 Start:
3     mov word    ptr loc_10106+1, 152h
4 loc_10106:   ;DATA XREF:
5     mov ax, 168h
6     mov word ptr loc_10129+5, ax
7 loc_10129:   ;DATA
8     mov word ptr es:0, 4D4Ch
```

Listing 4.1: Self-modifying code snippet

while single stepping inside a debugger on an `80486` *processor, what is the value of the 16-bit word at location* `loc_10129+5`*?* a) 168h, b) 152h, c) 4D4Ch, d) Value is unknown, e) None of the above.

The corresponding discussion first explores prefetch caches, as well as explaining more modern uses of the cache (ex: clearing the cache when a branch occurs). It also explains how a debugger handles self-modifying code while single stepping (in the context of clearing the cache on each step), and then uses a portion of the Intel IA-32 manual to explain how certain families handle self-modifying code. The peer instruction question can then be asked, and after the related discussion, the lecture can then turn to other methods of defeating debuggers.

Concept Triggers: Deconstructing this question, it allows for the usage of none of the above. Additionally, this question requires quite a bit of analysis–students must read and understand the code snippet as well as what the question is asking in order to better understand how it will ultimately execute in the particular situation–so this question falls under the category of analysis and reasoning questions.

Question Presentation: This question is clearly an example question, as it provides a code sample and requires that students answer by making inferences from the code sample as well as the question itself.

4.4 Network Penetration Testing

The problem of obtaining user credentials is central to most aspects of network penetration testing. Therefore, a non-trivial amount of time is spent on various password cracking techniques. The real point is, of course, is to gain an understanding of what methods can be used to thwart such attacks. A common point of misunderstanding is the purpose of salting password hashes, and what types of problems it can address. Salting is the prepending of random bits to the password prior to hashing.

Although the salt is not a secret, it can render efforts to reverse password hashes that rely on precomputed mappings computationally infeasible. At the same time, salting does little to prevent the cracking of weak passwords, as they can be effectively broken with dictionary techniques. An appropriate conceptual question that can lead to the various considerations is: *You obtain a leaked database of unsalted SHA-1 password hashes. What would be the most effective way to obtain as many passwords as possible in a short amount of time?* a) brute force, b) rainbow tables, c) dictionary attack with a large wordlist, d) passing the hash, and e) birthday attack.

The corresponding lesson first begins by discussing simple password guessing, noting that simple guessing is limited by speed as well as account lockouts. Then it moves to means of automation, pointing out that automation works best against collections of hashes, allowing for much greater speeds. Methods of password cracking such as rainbow tables and dictionary attacks can be discussed, leading to the conceptual peer instruction question to gauge understanding of the situational advantages. Following the question and discussion, the lecture can then turn to means of optimization of these approaches through heuristics, and then the instructor can then demonstrate common password cracking tools as well as provide an example hash for a hands-on lab.

Concept Triggers: Deconstructing this question, it is "qualitative", as it provides enough information about the password hashes to require students to identify concepts (hashing,

salting, etc.) and understand the relationship between the presence or stated lack (hash salting) of concepts to correctly answer the question, rather than a simple equation. Secondly, this requires students to "interpret representations"–they must identify the key words and concepts and make interpretations based on their usage. Finally, this question uses the "strategize only" question trigger, as students must identify the best path or tool to a solution rather than a solution itself.

Question Presentation: This question is clearly a scenario question, as the question presents a situation (acquisition of a hash dump).

Chapter 5

Analysis of Peer Instruction Questions

This chapter provides an analysis of 279 peer instruction questions recently developed for three cybersecurity courses: introduction to computer security (93 questions), network penetration testing (78 questions), and introduction to computer forensics (108 questions). The goal of the analysis is to identify the concept triggers (in Table 6.1) and presentation types in the questions of each of the three courses, and then, compare them.

For introduction to computer security, all 93 questions have multiple triggers. Excluding the "none of the above" trigger (as this is present in nearly all the questions due to the none/more than one of the above option), we have 84 questions with multiple triggers. Excluding the "identify a set or subset" trigger (as this is also present in nearly all the questions due to the same option), we have 83. Excluding both of those concept triggers, we have 56 questions with multiple triggers. For network penetration testing, all 78 have multiple triggers, due to questions with the qualitative question trigger. Finally, for Introduction to Computer Forensics, 67 of the 108 questions use multiple concept triggers. Excluding the "none of the above" trigger, there are 48 questions with multiple triggers, and excluding "identify a set or subset", there are 57 questions with multiple triggers.

5.1 Concept Triggers

Figure 5.1 presents the percentage of questions having concept triggers under consideration. There are a number of concept triggers conspicuously missing from both courses. This is because when building a question with particular concept triggers, there are many (such as "remove nonessentials") triggers that work well when developing a question, but are difficult to identify when dissecting a conceptual question; however, a number of concept

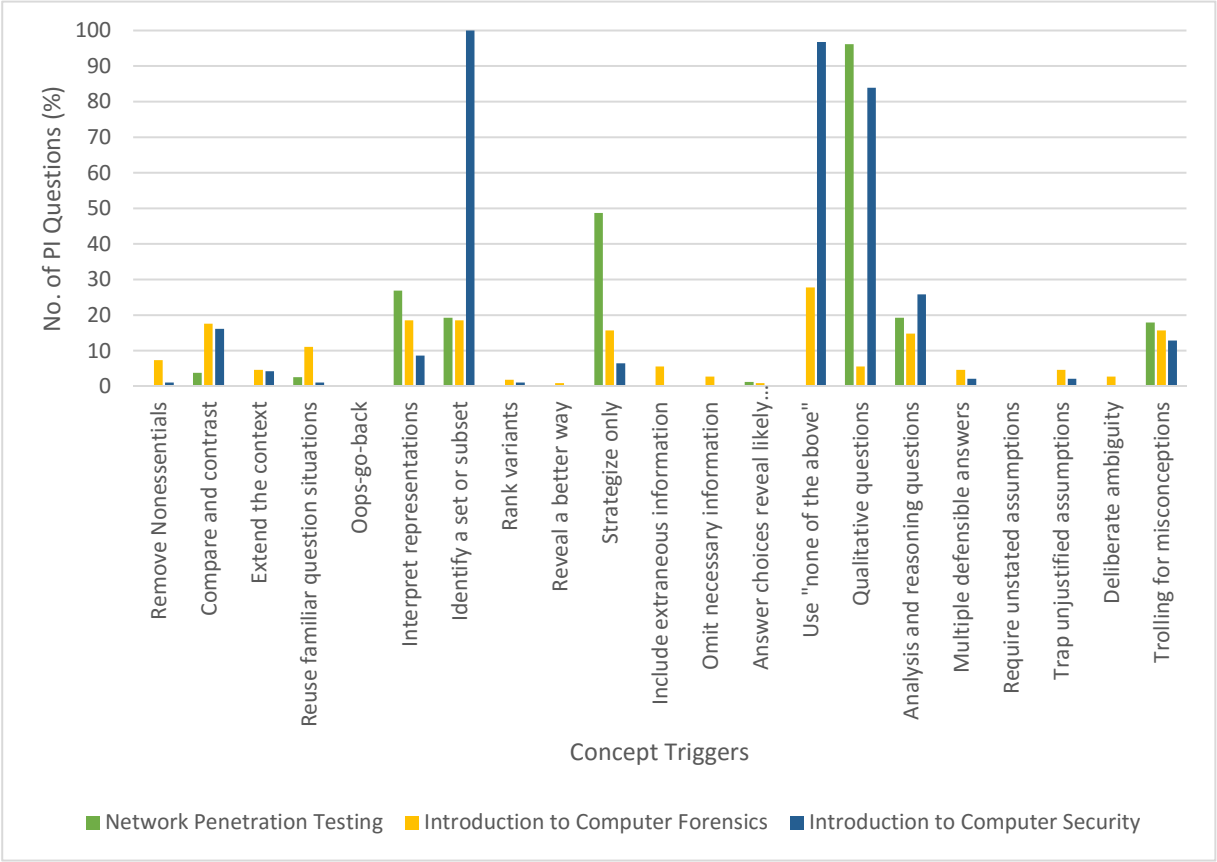


Figure 5.1: Percentage of peer instruction questions over concept triggers

triggers present in questions are in fact evident, and provide an interesting view into the creation of a peer instruction question set as a whole. While it may be difficult to identify items such as oops-go-back pairs, the ability to view trends of concept trigger usage shows interesting insight into the intentions of the instructor.

The three courses have a majority of questions that are qualitative. It can be rationalized, as the peer instruction questions are generally qualitative because they refer to concepts, relationships between concepts, etc. The third course, introduction to computer forensics, does not include a large number of questions specifically marked as qualitative as that trigger was not in itself specifically targeted as a concept trigger during development of that course's questions.

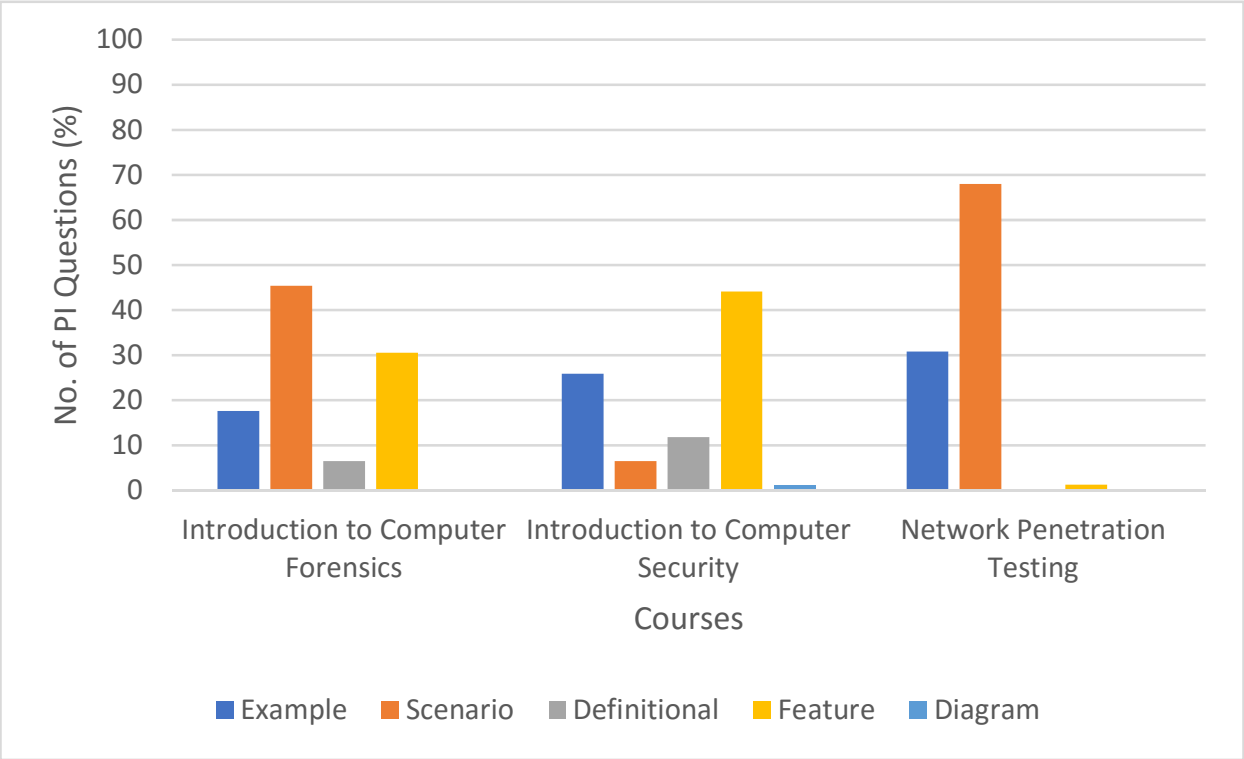


Figure 5.2: Percentage of peer instruction questions over presentation types

5.2 Question Presentations

We categorize the peer instruction questions based on the wording or presentation of the questions. The presentation types identified in our questions are scenario, example, definitional, diagram, and feature.

Figure 5.2 presents the percentage of peer instruction questions over presentation types. Both of the courses have a similar number of example questions. However, the introduction to computer security course has a significant number of feature questions. The network penetration testing course, on the other hand, has a majority of scenario-based questions.

There is no strict rule as to when to use each question presentation type; this is up to the instructor. However, it is important to consider the class material. For example, in

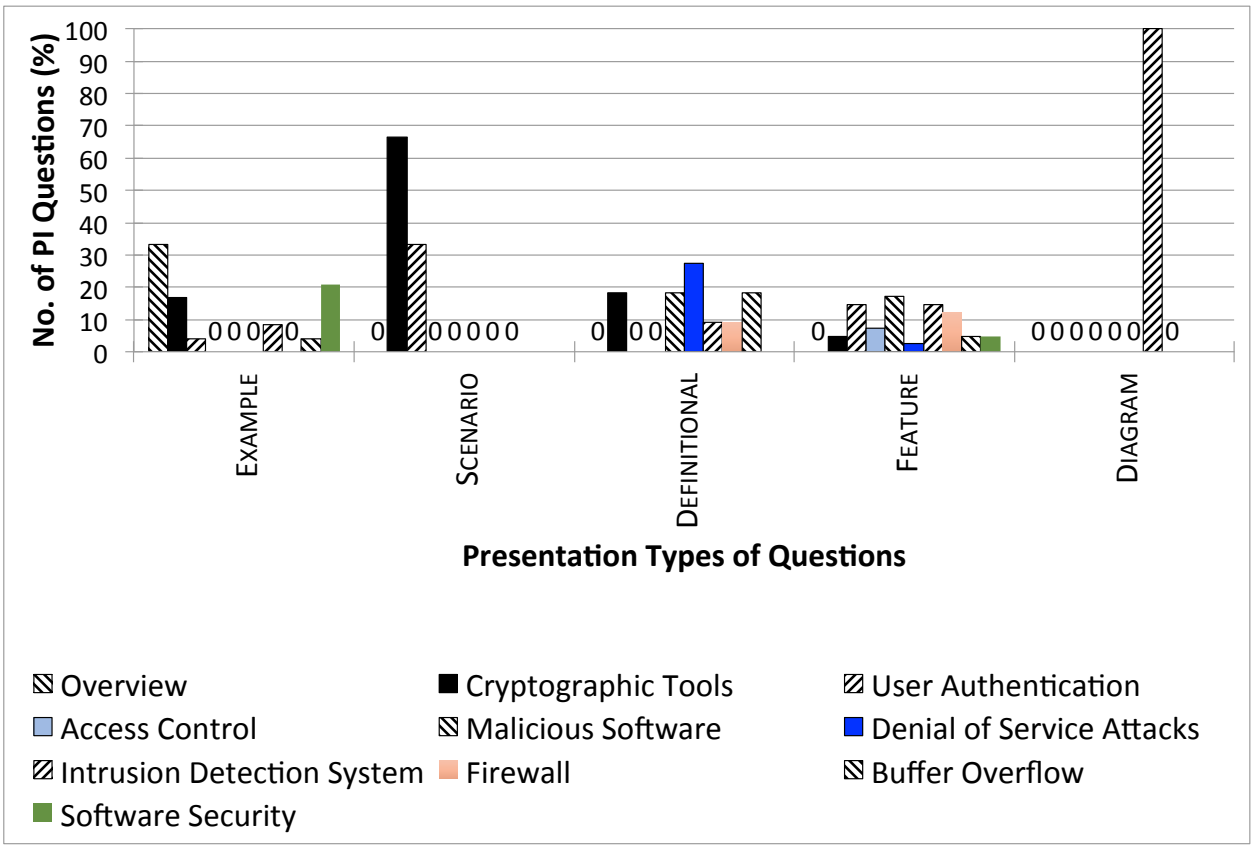


Figure 5.3: Percent distribution of questions as per presentation types and topics

our penetration-testing course, most of the material discussed lends itself well to hands-on activities, and much of the class time is used for hands on activities in a lab. For concepts that arise in practical material such as this, it would be helpful to trend toward using more scenario-based questions. When discussing subjects such as relationships between concepts or objects–for example, redirection of *stdin* and/or *stdout* through a *Netcat* instance–an instructor may find it useful to provide a diagram and focus the question around that. It is largely up to the instructor, but question presentation, much like question triggers, should be used to enhance peer discussion as deemed necessary.

We further analyze Figure 5.2's data to answer two questions: 1) what presentation types are used more frequently for different cybersecurity topics, and 2) what is the association between presentation types and concept triggers. Both aspects are critical for developing an effective peer instruction question.

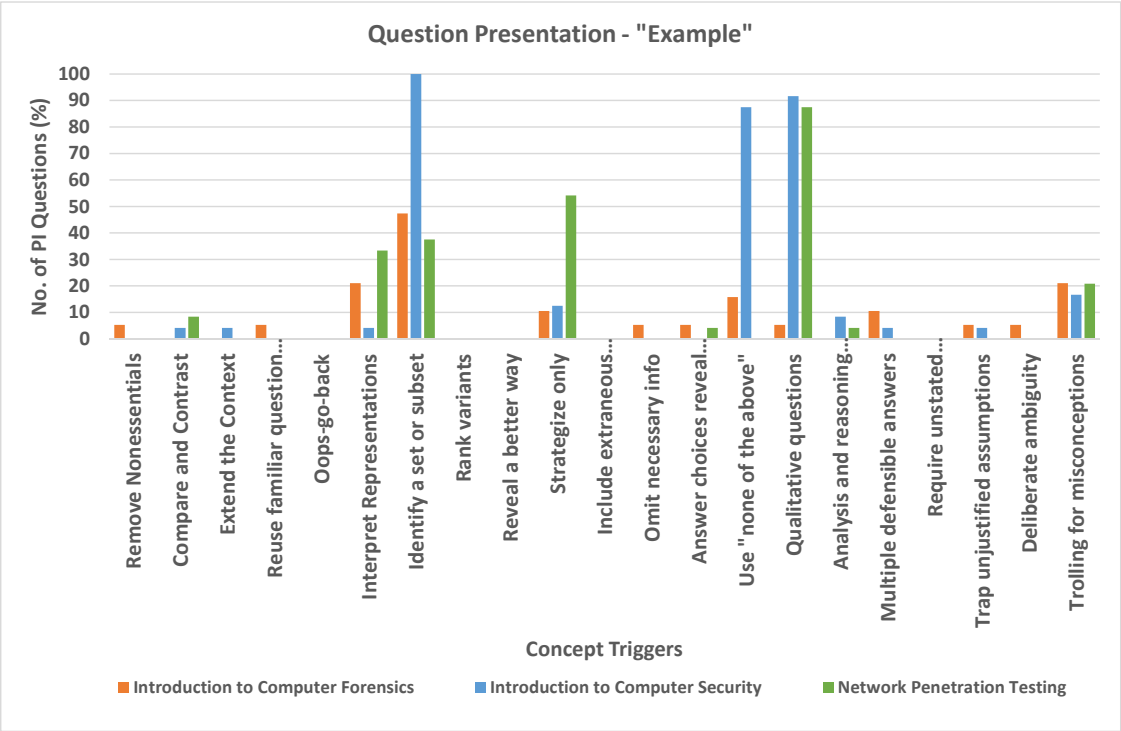
5.2.1 Presentation types vs. cybersecurity topics

To answer the first question, we have only considered the peer instruction questions for the course introduction to computer security. The course covers ten significantly different areas of cybersecurity (listed in Figure 5.3), and its questions are spread out across all the presentation types (refer to Figure 5.2). For the analysis, we further find the distribution of questions in accordance with cybersecurity topics and presentation types. Figure 5.3 presents the results showing that scenario based questions are from the topics, cryptographic tools, and user authentication. Apparently, these topics are traditionally discussed in scenario settings such as exchange of shared keys by Alice and Bob, or an attack scenario to steal and brute force a password file. Example, definitional, and feature based questions are spread out across the topics. Interestingly, the whole dataset contains only one diagram question, and that is for buffer overflow.

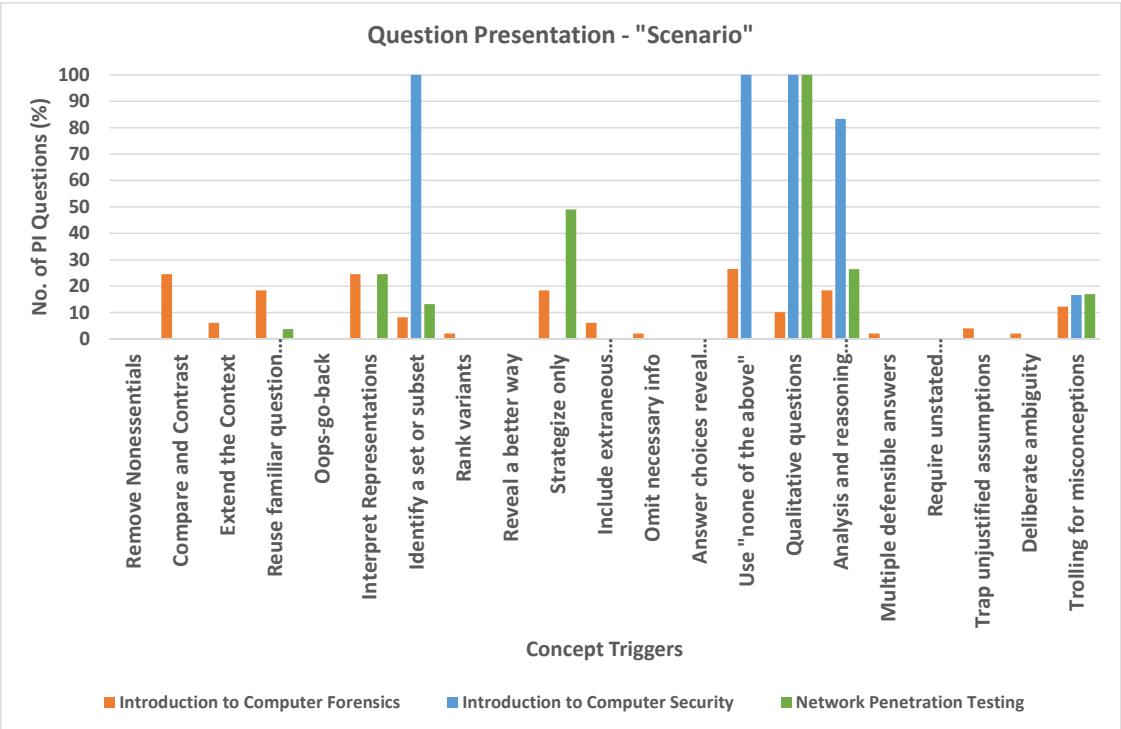
5.2.2 Association between presentation types and concept triggers

Figure 5.4 presents the results of the distribution of questions of a certain presentation type with respect to concept triggers. The analysis of the results shows that each course utilizes similar concept triggers for Example and Scenario-based questions. The exceptions are “identify a set or subset” and “use ’none of the above”’ that are used particularly heavily by the introduction to computer security, and “qualitative questions” that is used heavily by introduction to computer security and network penetration testing. Definitional and feature types of questions are mostly used for the introduction to computer security and network penetration testing courses. Interestingly, Compare and contrast is used in all three courses, but primarily for the ”feature” presentation type. Presentation type Diagram only has one question in our dataset for the course, introduction to computer security, utilizing three concept triggers: Interpret Representations, Identify a set or subset, and ”none of the

above". In general, Diagram questions are time-consuming and more difficult to create, and are therefore less likely to be popular for peer instruction questions.

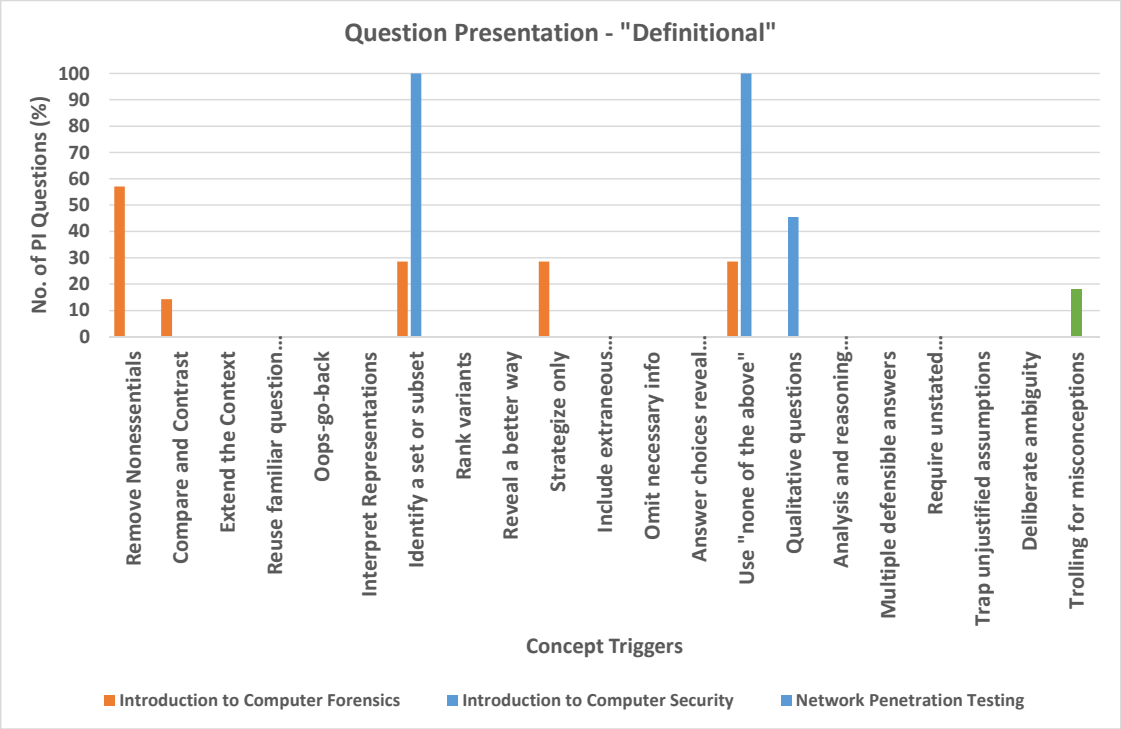


(a) Questions of presentation type Example are analyzed

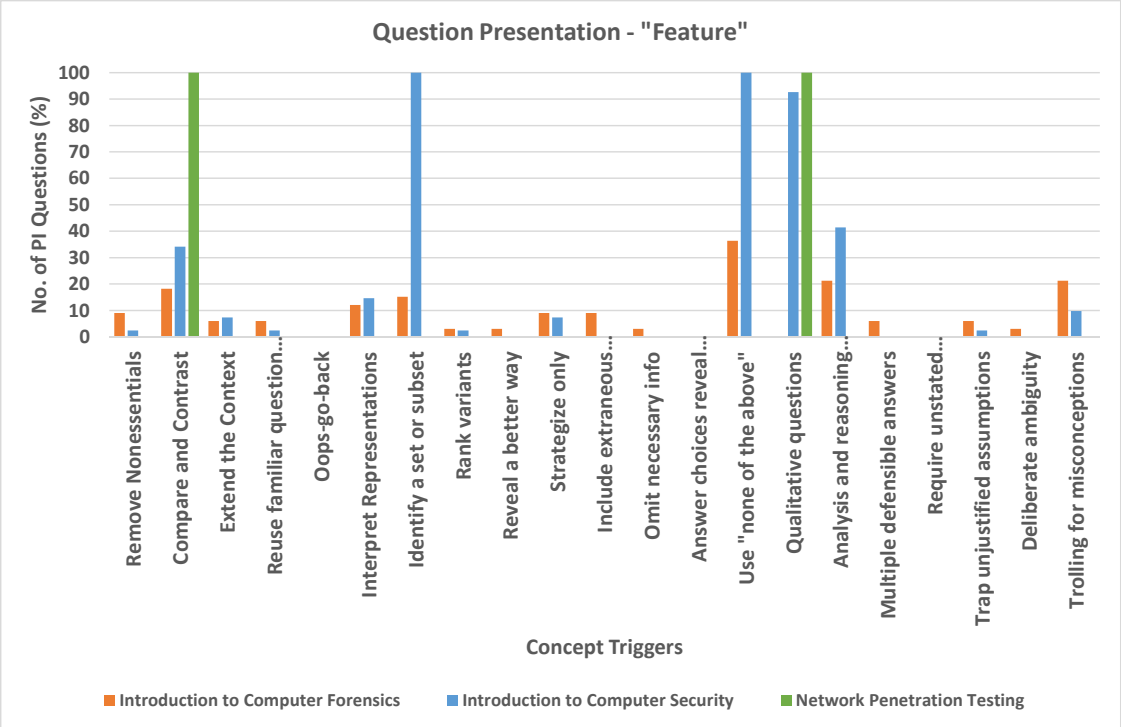


(b) Questions of presentation type Scenario are analyzed

Figure 5.4: Association between concept triggers and question presentations.



(c) Questions of presentation type Definitional are analyzed



(d) Questions of presentation type Feature are analyzed

Figure 5.4: Association between concept triggers and question presentations.

Chapter 6

Computer Forensics Workshop

6.1 Workshop Introduction

We chose to hold a small condensed workshop focusing on a few topics within the introduction to computer forensics course in order to gain experience implementing peer instruction within the classroom as well as gathering some initial data through the use of surveys on the material and peer instruction as well as quizzes preceding and following each of the topics to gain insight into learning progression through the workshop. In addition, the workshop allows us to gain insight into problems (misunderstanding the question or answers due to issues with wording, pacing of the workshop and/or peer instruction questions and discussion, etc.) that we may frequently see when developing questions.

We advertised the workshop approximately 10 days prior through an email advertising it as a short overview on important topics for computer forensic investigation as well as an introduction to usage of peer instruction, providing a signup form.

The workshop focused on a lecture with topics chosen from our Introduction to Computer Forensics course. It began with our survey on the material, interest and experience with cybersecurity and related fields, and some information on clicker questions. The lecture was interspersed with related peer instruction questions and quizzes, continued with a small hands-on component showing typical usage of Autopsy (computer forensics investigation software) and FTK Registry Viewer (software used to browse Windows registry contents), and concluded with a peer instruction and clicker usage survey. The following sections provide greater detail into the components of the workshop.

6.2 Advertisement and Student registration

The email that we sent to students as an advertisement contained a short form requesting contact information as well as some information on course level and willingness for preparation for the workshop. The registration form contains the following fields:

- Name
- Email address
- Have you taken CSCI 2125 (Data Structures)?
- Have you taken CSCI 4623/5623 (Introduction to Computer Forensics)?
- Have you taken CSCI 4401/5401 (Principles of Operating Systems)?
- Have you taken any cybersecurity courses?
- Are you willing to read some introductory material before the workshop?

We received 20 responses. 75% of the respondents had taken our data structures course. Only one respondent had previously taken introduction to computer forensics. 40% had taken principles of operating systems. 30% of respondents had taken cybersecurity courses previously. All respondents responded as willing to read introductory material.

6.3 Pre-class reading material

We provided two pieces of reading material. The first was a short paper detailing the importance, history, structure of, and useful data found in the Windows registry [4]. The second consisted of the first five pages of an article introducing file carving, touching on the FAT32 and NTFS file systems as well as their file allocation and deletion procedures, detailing fragmentation and means of file recovery in both, as well as carvers utilizing header and footer file magic [8].

In addition, we distributed a small quiz using Google Forms to respondents of the interest form to attempt to ensure better preparation for the workshop. The questions consisted of the following:

1. Which of these is not a Windows registry root key?
   (a) HKEY_LOCAL_MACHINE
   (b) HKEY_USERS
   (c) HKEY_CURRENT_CONFIG
   (d) HKEY_ROOT_USER
   (e) HKEY_CLASSES_ROOT
   Answer: HKEY_ROOT_USER

2. Which of the following utilizes a bitmap to denote cluster allocation?
   (a) FAT16
   (b) NTFS
   (c) FAT32
   Answer: NTFS

3. Which of these carvers was built as a direct improvement to Foremost?
   (a) Photorec
   (b) Scalpel
   (c) Binwalk
   (d) FTK carver
   (e) Magic Rescue
   Answer: Scalpel

4. Which of these features of the registry is most known for using ROT13 for encoding data?
   (a) Autorun
   (b) MRU
   (c) UserAssist
   (d) USBSTOR
   Answer: UserAssist

5. Which web browser utilizes the TypedURLs registry key?
   (a) Mozilla Firefox
   (b) Internet Explorer
   (c) Google Chrome
   (d) Opera Browser
   Answer: Internet Explorer

For the quiz, we received 11 responses. 7 of those answered all questions correctly, and 4 respondents answered with 3/4 questions correct.

6.4 In-class peer instruction activities

For the workshop peer instruction component, we had one group of four students, two groups of three, and one pair–each of these were groupings of students based upon where they chose to sit. Seating was restricted to the front of the room where clickers were placed with workstations. The lecture was centered around seven peer instruction questions, distributed among four separate lecture sections: "Introduction to Computer Forensics," "File Systems," "File Carving," and "Windows Registry." As we began each question, we provided the students 2-3 minutes to respond. 3-5 minutes was provided for them to discuss their answers amongst themselves, and we provided another 2-3 minutes for the second set of responses to the question.

6.4.1 Student participants

Of the 20 responses to the interest form, ultimately 12 attended. We provided a survey focused on digital forensics, security experience, opinion on the potential for peer instruction, and some prior testing information. 9 of the 12 attendees completed the survey, and Table 6.1 provides a short summary of some of their prior testing scores, while Table 6.2 discusses their coursework and experience in relevant fields. The full survey question set will be provided in Appendices A, B, and C.

Student	Degree level	GRE	ACT	SAT (1600)	SAT (2400)	High School GPA
1	Undergraduate				1800-2099	
2	Undergraduate		19-24			3.00-4.0
3	Undergraduate			1300-1600	1800-2099	3.00-4.0
4	Undergraduate		25-30			3.00-4.0
5	Undergraduate			1300-1600	1800-2099	3.00-4.0
6	Undergraduate				1800-2099	3.00-4.0
7	Undergraduate				1800-2099	3.00-4.0
8	Undergraduate		31-36			3.00-4.0
9	Undergraduate	300-320	19-24	1000-1299	1200-1499	1.00-1.99

Table 6.1: Testing background information for computer forensics workshop attendees

Student	Has taken security coursework	Intends to specialize in computer security at UNO	Intends to take more security courses	Has experience in
1	Yes	Yes	Yes	Systems adminstration, networking, OS internals, computer security
2	No	No	Yes	Networking, OS internals
3	No	Yes	Yes	Systems administration
4	No	Yes	Yes	Systems administration, networking
5	No	No	No	Networking, OS internals
6	Yes	Yes	Yes	Networking, OS internals, computer security
7	No	Yes	Yes	Networking, OS internals
8	No	No	Yes	
9	No	Yes	Yes	

Table 6.2: Coursework and experience related information for computer forensics workshop attendees

6.4.2 Peer Instruction Questions by Workshop Section

Here we summarize the four sections utilized in the workshop. We also include a listing of the seven peer instruction questions, their answers, and a short analysis of the concept triggers and presentation types.

TOPIC 1: *Introduction to Computer Forensics*

This section of the workshop serves as an introduction to computer forensics. We introduce various examples, sources, and benefits of digital evidence; a short summary of

the typical forensic investigative process; and we also discuss the difficulty of destroying digital evidence. These topics all lead into the following peer instruction question.

After the question is asked, we discuss the formatting process on Windows–particularly the difference between the traditional (or "quick") format, more thorough full formatting, and the importance of understanding formatting procedures on different devices and technologies.

This section's peer instruction question is as follows:

1. Estimate the fraction of disk blocks affected by formatting a hard disk.
   (a) 100%
   (b) 65%
   (c) 20%
   (d) Less than 5%

   Answer: *Less than 5%*

This question was previously discussed in Chapter 4, so I will simply list its concept triggers and presentation type.

Concept Triggers: This question uses "omit necessary information," "trap unjustified assumptions," "trolling for misconceptions", "deliberate ambiguity", and is a "qualitative question."

Question Presentation: This is a "feature" question.

TOPIC 2: *File Systems*

In this section, we introduce the FAT file system structure and its two primary data structures–directory entry and file allocation table. Without detail, we ask the first peer instruction in this section.

Following the first question, we detail the specifics of the two data structures, the means of data storage, details such as short filename and date/time storage in FAT12/16. Then we provide the primary differences between FAT12/16 and FAT32–the size of the FAT

entry, root directory size, and long filename support. Then, the second peer instruction question of this section is asked.

Following the second file system question, we discuss the regions of the FAT partition: the reserved region, the FAT region, the root directory region, and the file and directory data region. This discussion concludes the file systems section. The section's peer instruction questions are as follows:

1. In order to access a file from a FAT filesystem, what information is absolutely necessary?
   (a) Name and ending address of file content
   (b) Name, file size, and ending address of file content
   (c) Name and starting address of file content
   (d) File size and starting address of file content
   (e) None of the above

   Answer: *Name and starting address of file content*

Concept Triggers: Here we use "none of the above" as a trigger to provide the possibility.

We also use "trap unjustified assumptions" to see if any student does not fully understand that the FAT file system's data structures allow file clusters to be accessed similarly to a linked list, as a directory entry points to a file's first cluster, and the file allocation table helps identify the next cluster–we can follow this to the block marked as EOF; we then don't need the file size.

Question Presentation: This question is presented as a feature question as it draws upon and asks about core features of the FAT file system and its data structures.

1. If sector 0 is lost/damaged in FAT12/16, what problem does it cause?
   (a) The volume's sector, cluster, etc. sizes cannot be determined
   (b) The volume's maximum file sizes become unavailable
   (c) The number of file allocation tables becomes unknown
   (d) More than one of the above
   (e) None of the above

Answer: More than one of the above ("The volume's sector, cluster, etc. sizes cannot be determined" and "The number of file allocation tables becomes unknown")

Concept Triggers: This question uses "none of the above".

We also use "analysis and reasoning questions" as the students must understand the difference between FAT12/16 and 32–specifically that sector 0 is backed up in FAT32 but not 12/16, as well as what specifically is stored there and its implications for file storage.

Question Presentation: This is a feature question, targeting the difference in core features of FAT12/16 and 32.

TOPIC 3: *File Carving*

This section begins with its first peer instruction question. Following that, we describe the goal and traditional means of file carving (header/footer searches). We then ask the second peer instruction question.

Following the section's second question, we provide examples of common file format headers and footers. Then we provide visuals of typical carving situations with and without file fragmentation and damage (showing headers, footers, and related and unrelated file clusters).

With a concluding slide describing some of the main issues with file carving, we proceed to the Windows registry section. The peer instructions for the file carving section are as follows:

1. File carving is especially useful in which of the following one or more situations?
   (a) An operating system drive is examined as an external drive
   (b) Many potentially desired files have been deleted
   (c) Files have recently been defragmented
   (d) The file allocation table on a FAT file system has been corrupted
   (e) More than one of the above

   Answer: More than one of the above ("Many potentially desired files have been deleted" and "The file allocation table on a FAT file system has been corrupted")

Concept Triggers: This question uses "identify a set or subset", present in the "more than one of the above" option.

It also uses "interpret representations," as students must consider what in each answer would make file carving inherently warranted–the reasoning for each situation being a potential answer is something that must be inferred.

Question Presentation: This is an example question, particularly due to the choice of "interpret representations", as students must choose the correct example to answer the question.

1. File carving is the most effective in which one or more of the following scenarios?
   (a) Drive is highly fragmented
   (b) Drive is recently defragmented
   (c) System used to examine drive has low space
   (d) System used to examine drive has high space
   (e) More than one of the above

   Answer: System used to examine drive has high space

Concept Triggers: We are attempting to test the students' knowledge of the carving process as well as some of the major issues that complicate the process. Deconstructing this question, we note that two triggers are used. First, as we provide the potential for multiple question choices, we use "identify a set or subset".

Additionally, as students require an understanding of the downsides of both fragmentation as well as the defragmentation process (fragmentation is difficult to deal with in carving, but defragmentation can destroy useful unallocated space from previous files), we use "trolling for misconceptions", as the option "Drive is recently defragmented" presents a more ideal layout in terms of data contiguousness (and thus is an attractive option), but there is a potential for loss of important data.

Question Presentation: This question does not specifically present a scenario, but provides examples of potential carving situations on both a target drive and an investigator's machine. Thus, this question uses the "example" presentation.

TOPIC 4: *Windows Registry*

This section opens with its first peer instruction question. Following that, we discuss the structure of the Windows registry–including keys, values, and data as well as the primary registry hives and their backing files. Then, after discussing the registry's *LastWriteTime* timestamp, we ask the secton's second peer instruction question. Following that, we then show the attendees various useful locations in the registry, and conclude the section. The registry section's questions are as follows:

1. Which of the following actions is best described as an example of registry forensics?

   (a) An investigator uses Volatility to examine a file
   (b) An investigator uses LiME to access a memory dump
   (c) An investigator reviews the SAM hive to obtain password hashes
   (d) An investigator examines access timelines using FTK

   Answer: An investigator reviews the SAM hive to obtain password hashes

Concept Triggers: This is an "analysis and reasoning" question, as students must understand what each (at least the correct) tool is used for, and how the target data is obtained.

Question Presentation: This is a feature question, as students must understand the specific details behind each answer (tool usage scenarios as well as type of target data).

1. A USB drive with an unknown owner is found in a corporate setting. How might a forensic investigator typically determine whether that particular drive was plugged into any given Windows machine?

   (a) Examine all ntuser.dat files to determine if a user plugged it into the machine
   (b) Check the system registry file to see if it was plugged into that machine
   (c) Check the software registry file to see if it has been used by any particular piece of software
   (d) More than one of the above
   (e) None of the above

   Answer: Check the system registry file to see if it was plugged into that machine

Concept Triggers: This question first uses "none of the above" to point the students toward each answer as being potentially valid, while not drawing attention toward the specificity of the software or ntuser.dat options or generality of the system option. With multiple potentially valid options, if there is difficulty, the instructor can discuss the merit of each option in situations such as those described, or perhaps in a situation narrowing toward a particular user's and/or software's usage of the USB drive in question.

"Identify a set or subset" is used for much of the same reason–each is a potentially valid location for information on that USB drive, but for the purposes of tying the drive to the system itself, system is the more general and correct option, as if the drive had been utilized by software or a particular user, it would still be visible in USBSTOR or a similar location.

Question Presentation: This is a scenario question. It provides a particular situation that a forensic investigator may face, and it asks for the best means to proceed.

6.4.3 Data Collection Instruments

In addition to the previously detailed peer instruction questions, our data collection also consisted of reading quizzes that preceded and followed the latter three workshop sections to measure improvement through each lecture section–the preceding and following quizzes asked equivalent questions.

Available in Appendix A, the quizzes asked short and easy to answer questions such as:

1. Which of the following is a known issue with carving?
   (a) Fragmentation
   (b) Milestones
   (c) Unprintable bytes in headers

   Answer: *Fragmentation*

The ease of the questions is intended, as these questions are not conceptual, but they are used rather to determine how students retained knowledge from each of the lecture sections. The workshop quizzes are intended to serve the purpose that testing would serve in terms of providing data in a full-length course–of course the workshop attendees are not graded, and the quizzes immediately precede and follow the material (rather than being presented potentially months after the material is presented), but we believe the metrics from workshop quizzes should prove useful. In addition, the more immediate feedback through quizzes can help identify potential problems with peer instruction questions, lecture content, and lecture pacing.

We also provided two surveys to the workshop: a computer forensics interest/experience survey given at the start of the workshop, as well as a final survey focusing on student experience of peer instruction, created and used by Beth Simon and Leo Porter of the UC San Diego Computer Science and Engineering department, and provided by Simon, Porter, and Cynthia Lee of the Stanford Computer Science department.

The computer forensics interest/experience survey was used to collect information such as gender, enrollment level (undergraduate or graduate), gender, interest in computer security coursework and specialization, experience in related fields, some prior testing scores (ACT/SAT/GRE and high school GPA), and a number of opinion-based questions using a Likert scale discussing topics such as importance of computer science and security topic knowledge and learning styles. The Likert scale opinion answers are shown in Table 6.3.

The peer instruction survey gathered information on prior usage of clickers, workshop preparation (reading material and quiz), peer discussion, clicker usage, and lecture pacing. It uses a number of questions with a Likert scale to determine lecture preparation as well as opinions on peer instruction, clicker usage, and attentiveness; there are also a few additional opinion questions discussing details specific to this iteration of the workshop. We have included a table of average responses to the Likert questions (higher percentages indicate the more positive end of the Likert scale).

It is important to note that though 12 students attended the workshop, we only received 9 responses to the surveys. While there may not be enough data to draw particularly strong conclusions from these surveys, the content of these surveys is available in Appendices B and C.

6.4.4 Results and Evaluation of Student Performance

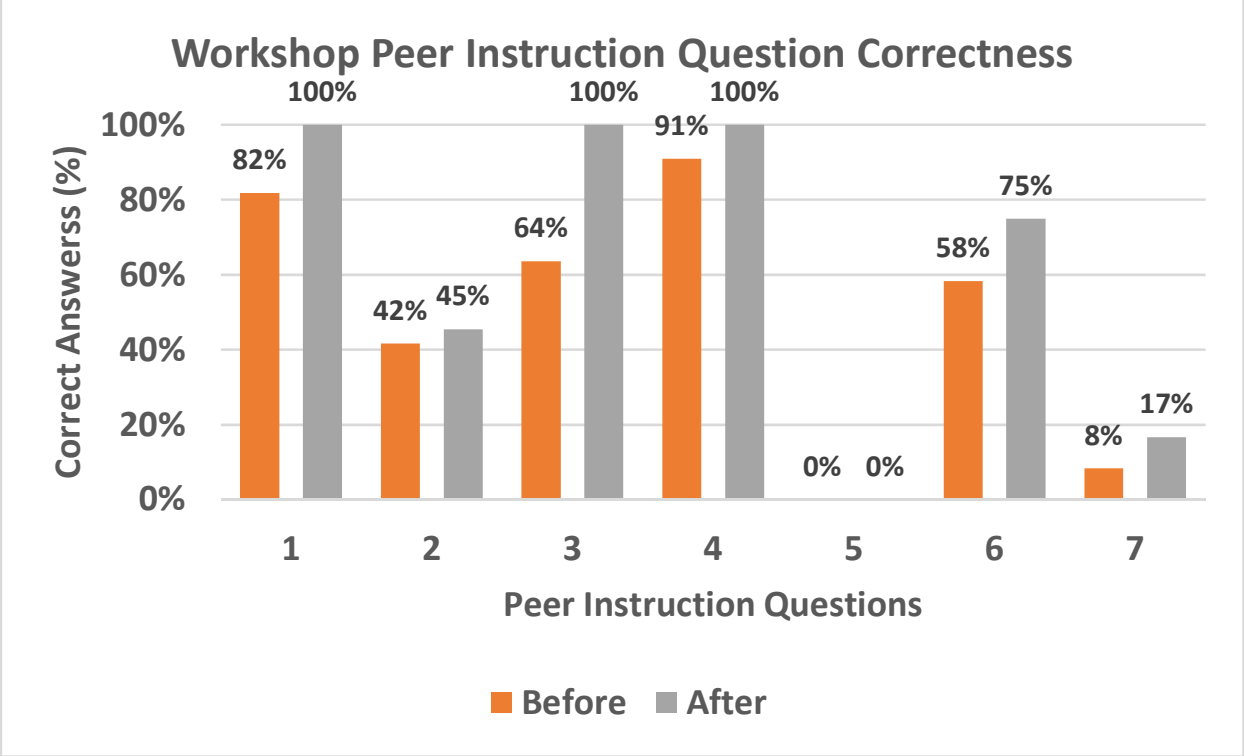


Figure 6.1: Percentage of correct answers to peer instruction questions during the computer forensics workshop

Answers to nearly every peer instruction question asked show a visible improvement, and keeping in mind that this gauges correct vs. incorrect answers in general and not the specific set of answers for each question, there was no deterioration in terms of performance across any of the question pairs. Figure 6.1 shows the improvements in correct answers across the entire set of question pairs.

In regards to the fifth peer instruction question used in the lecture (seen in Figure 6.1), we believe the failure to correctly answer it for either iteration may be due to the wording of the question or its answers.

There was feedback during the workshop that suggested some attendees may have felt that the question was a comparison of file carving vs. other unnamed evidence collection techniques rather than a comparison of potentially ideal usage situations for file carving. Additionally, we feel there was also confusion as to whether the particular answers used referred to the situation on an investigator's or a suspect's drive as well as "space" meaning "free space" in the case of answers c and d. We feel this is still a valuable question, and we have made corrections to the question wording.

We have modified this question and its answers to read:

1. In which of the following situations is file carving most effective?
   (a) The targeted drive is highly fragmented
   (b) The targeted drive has been recently defragmented
   (c) The system being used to examine the drive has low free space
   (d) The system being used to examine the drive has high free space
   (e) More than one of the above

   Answer: *The system being used to examine the drive has high free space*

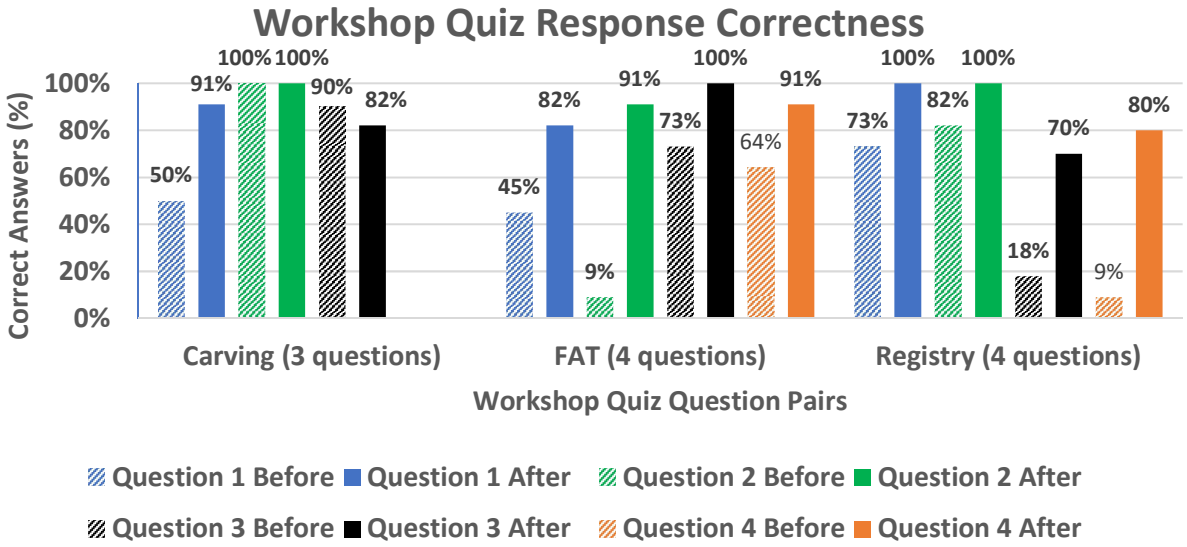


Figure 6.2: Percentage of correct answers to quiz questions per section during the computer forensics workshop; note that shaded bars followed by solid bars refer to the question's first and second iterations, respectively, for that section

Across the second iteration of every quiz, student performance showed a general improvement as seen in Figure 6.2. For these results, we note there were no major issues, and the improvements seen in the carving quiz question 1, FAT quiz question 2, and registry quiz questions 3 and 4 are particularly interesting to note.

Also, as in Tables 6.3, 6.4, and 6.5, the students had particularly positive opinions of peer instruction and clicker usage in general as well as computer security and forensics, and their workshop experience seems to have been relatively positive. Therefore, given both their in-workshop results and survey responses, we view peer instruction as an instructional method that we would like to adopt further.

Table 6.3: Computer forensics and cybersecurity interest and experience opinion survey

Question	Average Opinion
I believe it is valuable and helpful to learn challenging academic content by discussing these challenging topics with my fellow classmates.	82%
It is important for computer science students to understand malicious software, which is a program that is inserted into the system to compromise the data or availability.	84%
I believe the content presented in this workshop is relevant to my studies as a computer science student.	84%
It is important for computer science students to have thorough knowledge of filesystem internals for digital forensics.	87%
It is important to maintain chain of custody for digital evidence for forensic investigations.	76%
I learn topics well when I work through problems and discuss concepts with my peers.	82%
It is likely that I will take computer security courses after completing this workshop.	76%
As a computer science student, I should be aware of the state of data on a storage drive after a format operation.	78%
When the instructor asks questions during the workshop, it is helpful for my learning.	80%
It is important to recover the contents of volatile memory when a computer is seized for an investigation.	84%
I take interest when digital forensic investigations are highlighted in the news.	78%
I would be interested in an alternative lecture structure including more discussion and interaction with classmates.	80%
To understand computer forensics, I discuss it with friends and other students.	71%
I am not satisfied until I understand why something works the way it does.	80%
I study computer forensics to learn knowledge that will be useful in my life outside of school.	84%
Nearly everyone is capable of understanding computer forensics if they work at it.	73%

Table 6.4: Peer instruction lecture preparation, peer instruction, and clicker usage opinions

Question	**Average Opinion**
Thinking about clicker questions on my own, before discussing with people around me, helped me learn the workshop material.	87%
I read the required material before the workshop.	89%
The pre-workshop reading quiz helped me recognize what was difficult in the reading.	76%
Most of the time my group actually discussed the clicker question.	87%
Discussing course topics with my seatmates in the workshop helped me better understand the workshop material.	96%
The immediate feedback from clickers helped me focus on weaknesses in my understanding of the workshop material.	91%
Knowing the right answer is the only important part of the clicker question.	49%
Generally, by the time we finished with a question and discussion, I felt pretty clear about it.	80%
Clickers are an easy-to-use class collaboration tool.	89%
Clickers helped me pay attention in this workshop compared to traditional lectures.	82%
Using clickers with discussion is valuable for my learning.	80%
I recommend that other instructors use this approach (reading quizzes, clickers, in-class discussion) in their courses.	91%

Table 6.5: Workshop-specific opinions

From the point of helping me learn, the content of clicker questions was				
Much too hard 0%	Too hard 0%	OK 100%	Too easy 0%	Much too easy 0%
In general, the instructor gave us enough time to read and understand the questions before the first vote.				
No, far too little time 0%	No, too little time 0%	OK amount of time 89%	Yes, too much time 11%	Yes, far too much time 0%
Which of the following best describes your discussion practices in this group?				
I always discuss with the group around me, it helps me learn 78%	I always discuss with the group around me, I don't really learn, but I stay awake 0%	I sometimes discuss, it depends 22%	I rarely discuss, I don't think I get a lot out of it 0%	I rarely discuss, I'm too shy 0%
The amount of time generally allowed for peer discussion was				
Much too short 0%	Too short 11%	About right 89%	Too long 0%	Much too long 0%
In general, the time allowed for class-wide discussion (after the group vote) was				
Much too short 0%	Too short 11%	About right 89%	Too long 0%	Much too long 0%
In general, it was helpful for the instructor to begin class-wide discussion by having students give an explanation.				
N/A - The instructor rarely did this 11%	It's not helpful to hear other students' explanations 0%	It was helpful to hear other students' explanations 89%		
The professor explained the value of using clickers in this class.				
Not at all 0%	Somewhat, but I was still unclear why we were doing it 11%	Yes, they explained it well 67%	Yes, they explained it too much 22%	

Chapter 7

Related Work

There have been prior forays into the usage of peer instruction in the computer science classroom. Porter et al. performed a multi-instutional study of the usage of peer instruction in seven instructors' introductory programming courses [14]. Considering instructors' prior experience (or lack thereof) utilizing peer instruction, they primarily focus on student perception of peer instruction using measurements such as perceived question difficulty, question time allowed, discussion time allowed, content difficulty, and more. From surveys used, they note that at least 71% of students would recommend other instructors use peer instruction, and instructors viewed noticeable changes in classroom experience. Noting that one course had less than ideal survey results, Porter et al. note that in that case, a grade hinged on correctness in peer instruction responses, and many students felt that the value of peer instruction was not well-explained [14].

Similarly, Porter et al. conducted a measurement of peer instruction across multiple small liberal arts colleges to measure the effectiveness of peer instruction in smaller classes, using data from five instructors at three institutions [15]. The authors noticed normalized gains in the same range or above that of larger universities with students generally approving of the method and their performances.

Sarah Esper discusses an introduction of peer instruction to a software engineering course with 189 students [3]. Utilizing an interesting modification to the standard peer instruction process in which a clicker question is initially shown without answers and both the students and instructor propose potential answer choices with discussions of those answers (though the instructor does not mention whether an answer suggestion is correct or incorrect), which the author views as a way to teach problem solving "when there is no right answer" [3]. The author notes that, after the course, 72% of the students would recommend

the course instructor, with 28% not recommending due to reasons such as there not being clear correct answers or clicker questions being unclear [3].

Liao et al. created modeling practices for student outcome prediction in a twelve-week introductory computer science course to identify struggling students (described as the students scoring in the bottom 40% in final exams) utilizing peer instruction results to predict final exam scores through a linear regression model with approximately 70% accuracy [7].

Lee, Garcia, and Porter examine effectiveness of peer instruction in two upper-level computer science courses: Theory of Computation and Computer Architecture, finding average normalized learning gains of 39% [6].

In order to provide an overview of learning gains (defined here as the percentage increase in performance from individual to group peer instruction votes), from sources where available, in Table 7.1 we establish a listing of standard peer instruction implementations across multiple courses, highlighting the name of the course, the number of students in the section (potentially combined), and learning gains recorded from peer instruction data.

Table 7.1: Reported normalized learning gains from related studies [1]

Course	Enrollment	Learning Gains	Citation
Computer Architecture	Unknown	36%	[6]
Theory of Computation	Unknown	43%	[6]
CS1	19,18,32	43%,48%,26%	[15]
Computational Organization	10	40%	[15]
Operating Systems	9	64%	[15]
Theory of Computation	13	54%	[15]
CS1	Unknown	41%	[19]
CS1.5	Unknown	35%	[19]

None of the previously mentioned works cover peer instruction in cybersecurity. Therefore, we have found it worth our time not only to develop materials for cybersecurity courses, but also create a methodology that can be used to assist others in ensuring that questions they develop are truly conceptual.

[1] Note that comma separated values in enrollment and learning gain cells indicate multiple sections of the same course from the row's source–each item in the enrollment list corresponds to the same item in the learning gains list.

Chapter 8

Conclusion and Future Work

Through the use of peer instruction, we seek to build problem solving skills and technical aptitude in students who take advanced cybersecurity coursework. The expectation of student preparation prior to class and significant discussion during class significantly help students to have better understand of the content and better learning experience in class.

Our question development methodology for peer instruction allows instructors to systematically create questions and smoothly transition from lecture style format to peer instruction. The results of our analysis of 279 peer instruction questions (developed for three cybersecurity courses) conclude that the example and scenario based questions are more suitable for peer instruction questions. The concept trigger qualitative question generally applies to peer instruction questions. However, depending on the subject area in cybersecurity, the concept triggers may or may not be appropriate for the peer instruction questions. For instance, concept triggers identify a set or subset and strategize only are mostly suitable for the introduction to computer security course and network penetration-testing course, respectively.

8.1 Future Work

As part of the future work, we plan to utilize the peer instruction questions in their respective courses, and evaluate their overall efficacy in class, while holding further workshops to help gauge student opinions on peer instruction as well as question quality. Furthermore, a drawback for our peer instruction question creation methodology is that it is not particularly quantitative. To solve this, metrics will be collected for each peer instruction question. In the future, we will use these metrics to gauge effectiveness of each peer instruction question by its chosen concept triggers and presentation types in hopes of determining either an ideal

proportion for each question presentation type or an ideal measurement of which question triggers lend themselves better to any particular presentation type.

References

[1] I. Beatty, W. Gerace, W. Leonard, and R. Dufresne. Designing effective questions for classroom response system teaching. *American Association of Physics Teachers*, 74(1), 2006.

[2] C. H. Crouch and E. Mazur. Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69, 2001.

[3] Sarah Esper. A discussion on adopting peer instruction in a course focused on risk management. *J. Comput. Sci. Coll.*, 29(4):175–182, April 2014.

[4] Derrick J. Farmer. A forensic analysis of the windows registry.

[5] William E. Johnson, Allison Luzader, Irfan Ahmed, Vassil Roussev, Golden G. Richard III, and Cynthia B. Lee. Development of peer instruction questions for cybersecurity education. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)*, Austin, TX, 2016. USENIX Association.

[6] Cynthia Bailey Lee, Saturnino Garcia, and Leo Porter. Can peer instruction be effective in upper-division computer science courses? *Trans. Comput. Educ.*, 13(3):12:1–12:22, August 2013.

[7] Soohyun Nam Liao, Daniel Zingaro, Michael A. Laurenzano, William G. Griswold, and Leo Porter. Lightweight, early identification of at-risk cs1 students. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*, ICER '16, pages 123–131, New York, NY, USA, 2016. ACM.

[8] A. Pal and N. Memon. The evolution of file carving. *IEEE Signal Processing Magazine*, 26(2):59–71, March 2009.

[9] L. Porter, C. Bailey-Lee, and B. Simon. Halving fail rates using peer instruction: a study of four computer science courses. In *Proceedings of the 44th ACM technical symposium on Computer science education*, Denver, CO, March 2013.

[10] L. Porter, C. Bailey-Lee, B. Simon, Q. Cutts, and D. Zingaro. Experience report: a multi-classroom report on the value of peer instruction. In *Proceedings of the 16th Annual Conference on Innovation and Technology in Computer Science Education*, Darmstadt, Germany, June 2011.

[11] L. Porter, C. Bailey-Lee, B. Simon, Q. Cutts, and D. Zingaro. Peer instruction: do students really learn from peer discussion. In *Proceedings of the 7th Annual International Computing Education Research Workshop*, Providence, RI, August 2011.

[12] L. Porter and B. Simon. Retaining 18-30% more majors with a trio of instructional best practices in cs1. In *Proceedings of the 44th ACM technical symposium on Computer science education*, Denver, CO, March 2013.

[13] L. Porter and B. Simon. Retaining nearly one-third more majors with a trio of instructional best practices in cs1. In *Proceedings of the the Special Interest Group on Computer Science Education Technical Symposium*, 2013.

[14] Leo Porter, Dennis Bouvier, Quintin Cutts, Scott Grissom, Cynthia Lee, Robert McCartney, Daniel Zingaro, and Beth Simon. A multi-institutional study of peer instruction in introductory computing. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE '16, pages 358–363, New York, NY, USA, 2016. ACM.

[15] Leo Porter, Saturnino Garcia, John Glick, Andrew Matusiewicz, and Cynthia Taylor. Peer instruction in computer science at small liberal arts colleges. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '13, pages 129–134, New York, NY, USA, 2013. ACM.

[16] B. Simon and Q. Cutts. Peer instruction: a teaching method to foster deep understanding. *Communications of the ACM*, 55(2), 2012.

[17] B. Simon, M. Kohanfars, J. Lee, K. Tamayo, and Q. Cutts. Experience report: peer instruction in introductory computing. In *Proceedings of the 41st SIGCSE technical symposium on computer science education*, Milwaukee, WI, March 2010.

[18] B. Simon, J. Parris, and J. Spacco. How we teach impacts student learning: peer instruction vs. lecture in cs0. In *Proceedings of the 44th ACM technical symposium on Computer science education*, Denver, CO, March 2013.

[19] Beth Simon, Michael Kohanfars, Jeff Lee, Karen Tamayo, and Quintin Cutts. Experience report: Peer instruction in introductory computing. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, SIGCSE '10, pages 341–345, New York, NY, USA, 2010. ACM.

Appendices

A Workshop Quizzes

A.1 File Systems Quiz

1. What are the two primary data structures of FAT systems?

   (a) Directory entry, File allocation table

   (b) Cluster entry, File allocation table

   (c) File entry, Quick allocation table

   Answer: *Directory entry, File allocation table*

2. How is the filename "conf.ini" stored in a FAT file system that utilizes short filenames?

   (a) `conf.ini`

   (b) `CONF.INI`

   (c) `CONFINI`

   (d) `CONF INI`

   Answer: *`CONF INI`*

3. FAT32 maintains a backup BIOS Parameter Block.

   (a) True

   (b) False

   Answer: *True*

4. How does a FAT file system denote the end of a file?

   (a) The number of the final cluster equaling the stored number of clusters (-1 for zero indexing)

   (b) A FAT entry marked EOF

   (c) The final FAT entry for the file is marked "NULL"

   (d) Each file is allocated the same initial space, and the first "NULL" entry in the file's allocation table is the first cluster following the end of file

   Answer: *A FAT entry marked EOF*

A.2 File Carving Quiz

1. Traditional carving uses these to find potential files:

   (a) Known headers

   (b) Known filenames

   (c) Allocated clusters following blocks of unallocated clusters

   (d) Recovered file system metadata

   Answer: *Known headers*

2. Which of the following is a known issue with carving?

   (a) Fragmentation

   (b) Milestones

   (c) Unprintable bytes in headers

   Answer: *Fragmentation*

3. File carving could efficiently utilize distributed systems.

   (a) True

   (b) False

   Answer: *True*

A.3 Windows Registry Quiz

1. What are the primary registry files known as?

   (a) Hives

   (b) Keys

   (c) Values

   (d) Root files

   Answer: *Hives*

2. What is the timestamp given to any registry key?

   (a) LastWriteTime

   (b) LastReadTime

   (c) CreatedTime

   Answer: *LastWriteTime*

3. Which of the following stores data in the registry?

(a) Key

(b) Value

(c) Data

(d) Hive

Answer: *Data*

4. Where can user password hashes be found?

(a) SYSTEM

(b) SECURITY

(c) SOFTWARE

(d) SAM

(e) DEFAULT

Answer: *SAM*

B Computer Forensics Workshop Interest/Experience Survey

For section 1, please check the appropriate boxes to indicate your response.

1. What is your clicker number? It should be on the sticker on the back of the clicker, below the barcode. If you're unsure, please raise your hand.

2. You are a(n):
   (a) undergraduate student
   (b) graduate student

3. Gender
   (a) Male
   (b) Female
   (c) Other

4. Have you previously taken any coursework at UNO related to computer security?
   (a) Yes
   (b) No

5. Do you intend to specialize in the computer security field while at UNO?
   (a) Yes
   (b) No

6. Do you intend to take computer security courses after this workshop?
   (a) Yes
   (b) No

7. Do you have experience in any of the following items?
   (a) Systems Administration
   (b) Networking
   (c) Operating System Internals
   (d) Digital Forensics
   (e) Computer Security

For section 2, where applicable, please select your scores for each of the following.

8. GRE
    (a) 260-280
    (b) 280-300
    (c) 300-320
    (d) 320-340
9. ACT
    (a) 1-6
    (b) 7-12
    (c) 13-18
    (d) 19-24
    (e) 25-30
    (f) 31-36
10. SAT (1600 scale)
    (a) 400-699
    (b) 700-999
    (c) 1000-1299
    (d) 1300-1600
11. SAT (2400 scale)
    (a) 600-899
    (b) 900-1199
    (c) 1200-1499
    (d) 1500-1799
    (e) 1800-2099
    (f) 2100-2400
12. High School GPA
    (a) 0.00-0.99
    (b) 1.00-1.99
    (c) 2.00-2.99
    (d) 3.00-4.0

The following section includes a number of statements that may or may not describe your beliefs about learning computer forensics and the computer security field in general.

If you don't understand the statement, leave it blank. If you do understand but have no strong opinion, circle 3. Work quickly and don't over-elaborate the meaning of each statement.

1 represents "strongly disagree," 2 represents "somewhat disagree," 3 represents "neutral," 4 represents "somewhat agree," 5 represents "strongly agree'

13. I believe it is valuable and helpful to learn challenging academic content by discussing these challenging topics with my fellow classmates.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

14. It is important for computer science students to understand malicious software, which is a program that is inserted into the system to compromise the data or availability.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

15. I believe the content presented in this workshop is relevant to my studies as a computer science student.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

16. It is important for computer science students to have thorough knowledge of filesystem internals for digital forensics.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

17. It is important to maintain chain of custody for digital evidence for forensic investigations.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

18. I learn topics well when I work through problems and discuss concepts with my peers.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

19. It is likely that I will take computer security courses after completing this workshop.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

20. As a computer science student, I should be aware of the state of data on a storage drive after a format operation.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

21. When the instructor asks questions during the workshop, it is helpful for my learning.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

22. It is important to recover the contents of volatile memory when a computer is seized for an investigation.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

23. I take interest when digital forensic investigations are highlighted in the news.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

24. I would be interested in an alternative lecture structure including more discussion and interaction with classmates.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

25. To understand computer forensics, I discuss it with friends and other students.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

26. I am not satisfied until I understand why something works the way it does.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

27. I study computer forensics to learn knowledge that will be useful in my life outside of school.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

28. Nearly everyone is capable of understanding computer forensics if they work at it.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

C Workshop Peer Instruction and Clicker Survey

1. What is your clicker number? It should be on the sticker on the back of the clicker, below the barcode. If you're unsure, please raise your hand.

2. Select all statements which are true of you.

   (a) I have used clickers before in a computer science class at this school.

   (b) I have used clickers before in a physics class at this school.

   (c) I have used clickers before in a biology or chemistry class at this school.

   (d) I have used clickers before in a psychology class at this school.

   (e) I have used clickers before in some other class at this school.

   (f) I have used clickers at some other institution before.

3. If you have used clickers in another class at this school, tell us the instructor name (or, if you can't remember, the class number):

   Please rate your level of agreement with the following statements.

   If you don't understand the statement, leave it blank. If you do understand but have no strong opinion, circle 3. Work quickly and don't over-elaborate the meaning of each statement.

   1 represents "strongly disagree", 2 represents "somewhat disagree", 3 represents "neutral", 4 represents "somewhat agree", 5 represents "strongly agree"

4. Thinking about clicker questions on my own, before discussing with people around me, helped me learn the workshop material.

   (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

5. I read the required material before the workshop.

   (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

6. The pre-workshop reading quiz helped me recognize what was difficult in the reading.

   (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

7. Most of the time my group actually discussed the clicker question.

   (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

8. Discussing course topics with my seatmates in the workshop helped me better understand the workshop material.

   (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

9. The immediate feedback from clickers helped me focus on weaknesses in my understanding of the workshop material.

   (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

10. Knowing the right answer is the only important part of the clicker question.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

11. Generally, by the time we finished with a question and discussion, I felt pretty clear about it.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

12. Clickers are an easy-to-use class collaboration tool.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

13. Clickers helped me pay attention in this workshop compared to traditional lectures.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

14. Using clickers with discussion is valuable for my learning.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

15. I recommend that other instructors use this approach (reading quizzes, clickers, in-class discussion) in their courses.

    (1) Strongly Disagree (2) Somewhat Disagree (3) Neutral (4) Somewhat Agree (5) Strongly Agree

16. Comments?

Please select the answers with which you agree most.

17. From the point of helping me learn, the content of clicker questions was

    (a) Much too hard
    (b) Too hard
    (c) OK
    (d) Too easy
    (e) Much too easy

18. In general, the instructor gave us enough time to read and understand the questions before the first vote.

    (a) No, far too little time
    (b) No, too little time
    (c) OK amount of time
    (d) Yes, too much time
    (e) Yes, far too much time

19. Which of the following best describes your discussion practices in this workshop?

    (a) I always discuss with the group around me, it helps me learn
    (b) I always discuss with the group around me, I don't really learn, but I stay awake
    (c) I sometimes discuss, it depends
    (d) I rarely discuss, I don't think I get a lot out of it
    (e) I rarely discuss, I'm too shy

20. The amount of time generally allowed for peer discussion was

    (a) Much too short
    (b) Too short
    (c) About right
    (d) Too long
    (e) Much too long

21. In general, the time allowed for class-wide discussion (after the group vote) was

    (a) Much too short
    (b) Too short
    (c) About right
    (d) Too long
    (e) Much too long

22. In general, it was helpful for the instructor to begin class-wide discussion by having students give an explanation.

    (a) N/A - The instructor rarely did this
    (b) It's not helpful to hear other students' explanations
    (c) It was helpful to hear other students' explanations

23. The professor explained the value of using clickers in this class.

    (a) Not at all
    (b) Somewhat, but I was still unclear why we were doing it
    (c) Yes, they explained it well
    (d) Yes, they explained it too much

Vita

William Johnson was born in Memphis, Tennessee. He obtained a B.A. in Sociology/Anthropology from Millsaps College with a concentration in Anthropology and began attending the University of New Orleans a short time thereafter with the intent of obtaining a M.S. degree in Computer Science under the Information Assurance concentration.