March 2017

Peer Review in Cybersecurity Education

William M. Temple
Worcester Polytechnic Institute

WPI

Peer Review in Cybersecurity Education

A Major Qualifying Project Report:

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

in

Computer Science

by

William Temple

Date: March 12, 2017

APPROVED:

Professor Kathryn Fisler, Major Project Adviser

Abstract

Demands for a well-prepared cybersecurity workforce are growing, and instructors who teach cybersecurity to students require effective tools and techniques. Peer review is one technique that has been demonstrated to have practical benefits in many contexts, including instruction. In this paper, we explore the use of peer review in two cybersecurity courses at WPI, and we analyze how students' reviews deal with the topic of cybersecurity. We find that while they utilize peer review in different ways, the two courses have similarities in their review texts. The construction of review prompts and other factors may influence the degree to which students focus on security. Finally, we discuss our findings and present suggestions for instructors who use peer review in cybersecurity courses.

Acknowledgments

I thank my adviser, Professor Kathryn Fisler, as well as Francisco Castro, both of whom provided invaluable assistance and insights formulating and investigating this topic. I thank Professor Craig Shue and Professor Krishna Venkatasubramanian for collecting and providing anonymized data from their courses for the purposes of this study. This research was supported by the National Science Foundation under grant number DGE-1500039.

Contents

List of Figures

1 Introduction

In Computer Science, much attention has been given to the improvement of pedagogy. Educators have explored a variety of programming languages, instructional models, and techniques in order to improve learning outcomes both within the classroom and beyond. Peer review, widely practiced in both academic and industrial settings, provides one well-studied mechanism for improving learning [1, 2, 3, 4, 5, 6, 7]. Peer review requires that students not only practice their own development skills, but also develop their ability to critically evaluate others' work. In addition to enhancing the learning experience, peer review also aids in the development of skills for industrial code review, such as the kind practiced in many large software development settings [8].

Also of much interest in Computer Science, the field of Computer Security (or cybersecurity) continues to widen, growing increasingly relevant to all aspects of professional programming practice. As the need for cybersecurity-educated personnel expands, we must give extra attention to the methods and practices that we use to teach cybersecurity. Industrial cybersecurity practice often demands that practitioners perform focused code-review and synthesize feedback from their colleagues with multiple perspectives. Peer-review, which incorporates the use of many of these same skills, therefore seems like a particularly good match for cybersecurity instruction.

In this paper, we study the use of peer review in two cybersecurity courses at Worcester Polytechnic Institute. We analyzed the reviews produced by the students, approaching the problem as an exploration of how the courses' different peer review configurations may affect students' tendencies to offer cybersecurity-related feedback. After constructing an initial framework for understanding students' commentary, we posed the following questions:

- Do reviewers choose to comment more on security characteristics or on other aspects of a submission?
- When reviewers choose to comment on security, do they do so in a concrete and

actionable manner, or in an abstract manner?

- How do the configuration parameters of the peer review environment relate to students' tendencies to provide security-related feedback?
- Do reviews by the same author remain consistent (in terms of security-orientation) across multiple sequential assignments?

These questions guide our exploration of the two courses' implementations of peer review. As a result, we develop a set of recommendations which instructors might use to enhance the use of peer review in the cybersecurity classroom.

2 Related Work

This paper discusses peer review in cybersecurity education. The topics of peer review in higher education (both generally and in Computer Science), of peer code review in cybersecurity professions, and of cybersecurity education, have all independently received ample coverage. However, we are aware of no prior research which integrates all of these topics. This section focuses on those three components of this research.

2.1 Peer Review in Higher Education and Computer Science

Topping, in his seminal work on peer assessment in education, describes the practice of peer review as "learning by assessing"[1]. He underlines that peer review is not conducted solely for the benefit of the student whose work is *assessed*, but also for the student who is the *assessor*. Lundstrom and Baker further found that, when using peer review in beginner writing classrooms, the assessor appeared to learn more after writing a review than the reviewee did after receiving it [2]. Dochy *et al.* discussed the benefits and drawbacks of peer assessment compared to traditional expert-based assessment. They found that peer assessment aided students in the formative stages of the learning process—specifically by requiring them to

compare different solutions to an assignment [3]. We are therefore interested to know if (and to what degree) students' reviews deal explicitly with cybersecurity material, as this body of existing work (and the diversity of educational domains under study) suggests that an engagement with cybersecurity in peer review may help students develop and solidify their knowledge of the topic.

Nelson and Schunn developed a method for qualitatively analyzing review contents in writing classrooms [9]. They analyzed pieces of feedback and classified statements using a data coding rubric which placed more value on specific or actionable comments rather than nonspecific comments. They also found a significant relationship between specificity and likelihood for the reviewers' suggestions to be implemented. We therefore seek to understand how students employ concrete and abstract feedback in their reviews and whether the peer review configuration may have an effect on the distribution of abstract and concrete commentary.

2.1.1 In Computer Science

A 2001 paper by Gehringer discusses the usefulness of peer review for students of various skill levels, "from second-semester programming to graduate reading courses"[4]. He also describes a web-based platform for peer review similar to the systems that are used at WPI and attests to students' increased familiarity with problems after completing reviews [4]. The utility of web-based peer review systems is expanded by Politz *et al.* [5], and further by Gehringer [10]. We are curious how differences in our systems (which are web-based), might effect how students respond to prompts.

Hicks *et al.* analyzed the effects of numeric rating prompts on reviewers' feedback. They found that the presence of numeric ratings in review prompts is correlated both with more explanatory (defined as a "suggestion justified with an explicit explanation") reviews and with more positive reviews [11]. However, they express concerns about the self-selected nature of their sample. Our data includes one set of reviews with numeric ratings of the entire

assignment and another set with numeric ratings of individual components of the assignment, so differences in positivity or concreteness in our data sets *may* be accounted for partially by this effect.

2.1.2 In-Flow Peer Review

Politz *et al.* define *In-Flow* Peer Review (IFPR) as a peer review done *in stages* while an assignment is in progress [5]. In an ITiCSE Working Group paper, Clarke *et al.* describe some goals of IFPR, ranging from metacognitive skills to fostering socialization, and they outline the potential of review prompts to focus reviewers on specific elements of a submission [6]. As one of the courses we studied utilized very specific prompts, and the other utilized general prompts, we are interested to know whether reviewers' focus on cybersecurity might be directed by the prompts.

In a 2014 study, Politz *et al.* examine peer reviews (conducted *in-flow*) of tests in tests-first programming using a manual qualitative coding methodology. They found that reviewers engage with course material thoughtfully while writing reviews, but noted that prior rubrics which "value targeted comments" may not apply as well to reviews which prompt the reviewer to identify content which is *missing* from a submission [7]. Both of the courses we studied in this article prompted reviewers to identify missing or incomplete information (in the form of unidentified vulnerabilities or exploits) in the submission artifacts that they review, so this distinction may prove important.

2.2 Cybersecurity Education

Demands for cybersecurity education continue to grow [12]. In general, the consensus on cybersecurity education seems to be that it demands active, rather than passive learning [13, 14, 15, 16, 17, 18]. Numerous works by several different authors describe "hands-on" laboratory environments for cybersecurity (also, Information Assurance[1]) education.

[1]NIST (via. Cooper *et al.*) defines Information Assurance as "a set of technical and managerial controls designed to ensure the confidentiality, possession of control, integrity, authenticity, availability, and utility of

Manson and Pike argue in their 2014 *ACM Inroads* article that "developing and measuring cybersecurity skills cannot be accomplished through traditional academic methods alone, there must be support for students to work independently and in teams"[18]. They further assert that students' foremost need in a cybersecurity classroom is hours of practice. We reason that the additional exposure to cybersecurity through peer review provides additional practice in an individual setting as well as a mechanism for assessing the reviewers' engagement with the material.

Vaughn *et al.* describe an information security curriculum which both integrates into existing coursework (beginning at the intermediate level) and expands into new courses [19]. This focus on early integration seems to corroborate the idea that the most significant force in cybersecurity education is time. We are therefore eager to know whether or not students are engaging in a discourse around cybersecurity in their peer reviews, as the act of reviewing may provide valuable cognitive reinforcement of cybersecurity knowledge.

2.3 Industrial Code Review

Peer review has become a very common practice in industrial and open-source programming. Cohen estimates that some industrial peer review practices may save a company up to 50% of overall development cost [8]. The MITRE Corporation extends their support of code reviews to include extra reviews "with a focus solely on security"[20]. Students in cybersecurity courses that employ peer review, therefore, are participating in an activity (evaluating peer work) which is commonplace in the work environment. We are additionally interested to know if students believe that peer review is a useful exercise, given its practical use.

Bachelli and Bird described the motivations for code review. They found that the among the most prominent motivations (based on coded responses to survey results) for code review was knowledge transfer [21]. Clarke *et al.* describe knowledge transfer as a goal of peer review (and of in-flow peer review specifically) [6]. This similarity seems a compelling reason to

information and information systems"[17].

consider the use of peer review in the cybersecurity classroom.

3 Data Gathering

We collected anonymized data from the following two cybersecurity courses:

- CS4401: *Software Security Engineering*, a senior-level course involving the analysis of software-level vulnerabilities in isolated systems (assumes familiarity with operating systems, C, UNIX, databases, and technologies for building web applications such as JavaScript)
- CS4404: *Computer Network Security*, a senior-level course in which students analyze the security of networked systems and networking infrastructure (assumes familiarity with operating systems, computer networks, and Linux or UNIX)

Specifically, we collected assignment submissions, peer reviews on those assignments, and reviews-of-reviews (meta-reviews). The *Software Security Engineering* instructor also provided some expert (course staff) feedback (including anonymized assignment and final grades). In this section, we compare the peer review implementations, submission artifacts, and review prompts of each course.

3.1 Peer Review Implementations

The two courses from which we gathered data did not use the same peer review procedures. Students in *Software Security Engineering* used CaptainTeach [22], an online peer review system. The CaptainTeach interface (Figure 1) presents the artifact under review next to a series of free-form and Likert-scale questions about the work. The reviews for this course were conducted in-flow. Students in this course were given the opportunity to modify their submissions based on review feedback before the final submission was due. The authors then

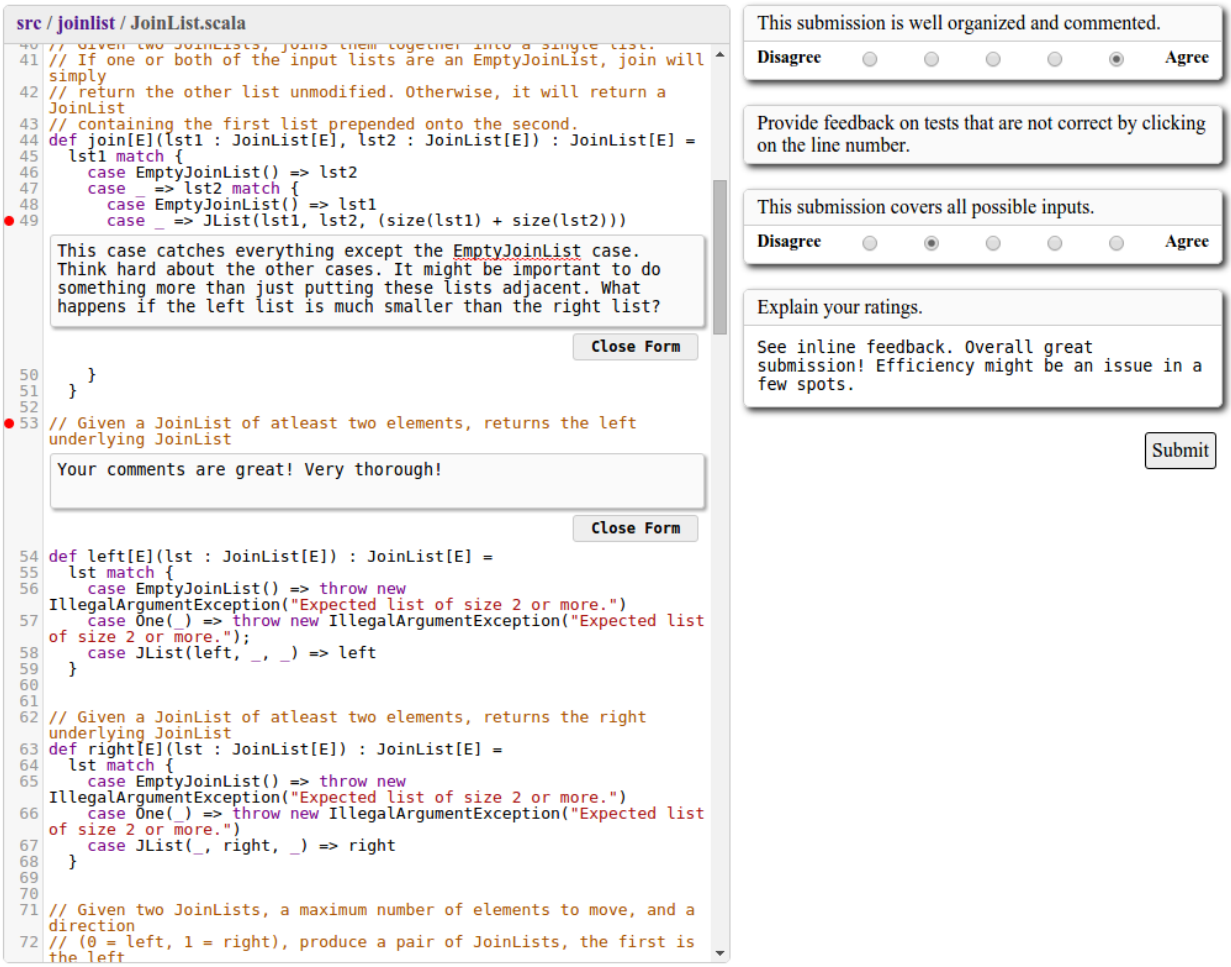
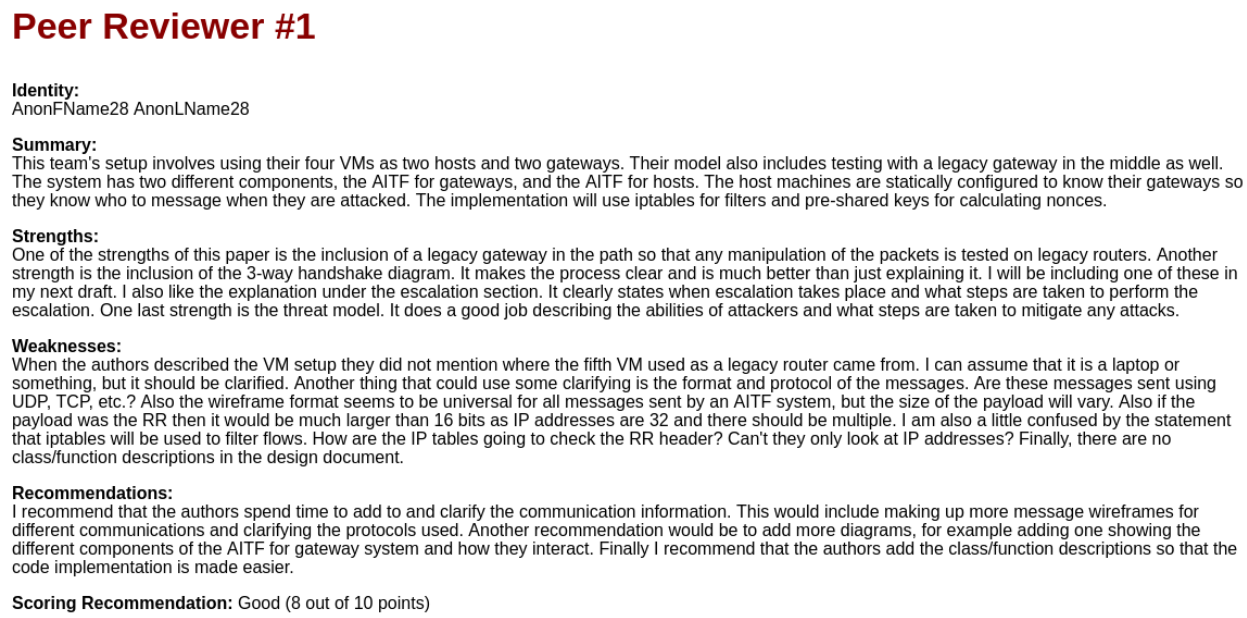


Figure 1: The CaptainTeach review interface with example code.

had the opportunity to respond to the review. The intermediate submission, which was peer reviewed, was not evaluated by the course staff.

Computer Network Security did not use an in-flow model, as the pacing of the course did not permit students to see the reviews until after the final submission's deadline. In this course, students used InstructAssist, a platform developed by the course's instructor that integrates an online peer review system. The InstructAssist interface (Figure 2) provides the reviewer with a series of general prompts about the artifact under review. The reviewers then received meta-reviews provided by the course staff. These meta-reviews included a short comment and a score on a linear scale out of 5 possible points.



Peer Reviewer #1

Identity:
AnonFName28 AnonLName28

Summary:
This team's setup involves using their four VMs as two hosts and two gateways. Their model also includes testing with a legacy gateway in the middle as well. The system has two different components, the AITF for gateways, and the AITF for hosts. The host machines are statically configured to know their gateways so they know who to message when they are attacked. The implementation will use iptables for filters and pre-shared keys for calculating nonces.

Strengths:
One of the strengths of this paper is the inclusion of a legacy gateway in the path so that any manipulation of the packets is tested on legacy routers. Another strength is the inclusion of the 3-way handshake diagram. It makes the process clear and is much better than just explaining it. I will be including one of these in my next draft. I also like the explanation under the escalation section. It clearly states when escalation takes place and what steps are taken to perform the escalation. One last strength is the threat model. It does a good job describing the abilities of attackers and what steps are taken to mitigate any attacks.

Weaknesses:
When the authors described the VM setup they did not mention where the fifth VM used as a legacy router came from. I can assume that it is a laptop or something, but it should be clarified. Another thing that could use some clarifying is the format and protocol of the messages. Are these messages sent using UDP, TCP, etc.? Also the wireframe format seems to be universal for all messages sent by an AITF system, but the size of the payload will vary. Also if the payload was the RR then it would be much larger than 16 bits as IP addresses are 32 and there should be multiple. I am also a little confused by the statement that iptables will be used to filter flows. How are the IP tables going to check the RR header? Can't they only look at IP addresses? Finally, there are no class/function descriptions in the design document.

Recommendations:
I recommend that the authors spend time to add to and clarify the communication information. This would include making up more message wireframes for different communications and clarifying the protocols used. Another recommendation would be to add more diagrams, for example adding one showing the different components of the AITF for gateway system and how they interact. Finally I recommend that the authors add the class/function descriptions so that the code implementation is made easier.

Scoring Recommendation: Good (8 out of 10 points)

Figure 2: The InstructAssist review interface with a sample review

3.1.1 Submission Artifacts

Significant differences in the types of artifacts that students produced in these two courses also add dimensionality to our analysis. In *Software Security Engineering*, the students analyzed a software system and produced lists of security vulnerabilities with a defined structure. They were asked to describe the vulnerability and provide instructions for exploiting it. This format afforded little flexibility in the structure of the submissions.

Conversely, *Computer Network Security* students studied and designed an implementation of Active Internet Traffic Filtering, "a scalable network-layer defense against internet bandwidth-flooding" described by Argyraki and Cheriton in 2009 [23]. The students produced multi-page text documents describing their designs, testing strategies, and test results. These artifacts did not require any particular structure.

3.1.2 Review Prompts

The courses used different sets of questions to prompt the reviewer. In *Computer Network Security*, students were given four general prompts, in which they were asked to (1) summarize

the document they reviewed, (2) describe its strengths, (3) describe its weaknesses, and (4) suggest improvements. After answering the prompts, the reviewers were required to give a grading recommendation on a linear scale from 0 (No Credit) to 10 (Superior).

Software Security Engineering, by contrast, used a more specific set of prompts. Reviewers were asked to rate the following statements on a Likert-scale (with a minimum score of 1 and a maximum score of 5) according to whether or not they agreed:

- These exploits correctly take advantage of the identified vulnerability.
- These exploits are qualitatively different from each other.
- This strategy adopted is systematic.

The reviewers were then provided a small free-form text box below each of the Likert-scale questions to explain their agreement or disagreement with the each statement. Finally, the reviewers were asked (using a free-form text box) to "Describe something [they] liked about these exploits."

3.2 Online Survey

We attempted to gauge students' sentiments on peer review by conducting an online survey of *Computer Network Security* students immediately following the conclusion of an active section of the course in Fall 2016. We asked questions (Appendix A) about the students' preferences regarding peer review styles, the perceived usefulness of peer review in the classroom, and gave the students an additional opportunity to voice their opinions. Survey participation was incentivized by means of a raffle ticket[2].

[2]this study and the collection of survey data was approved by WPI's Institutional Review Board

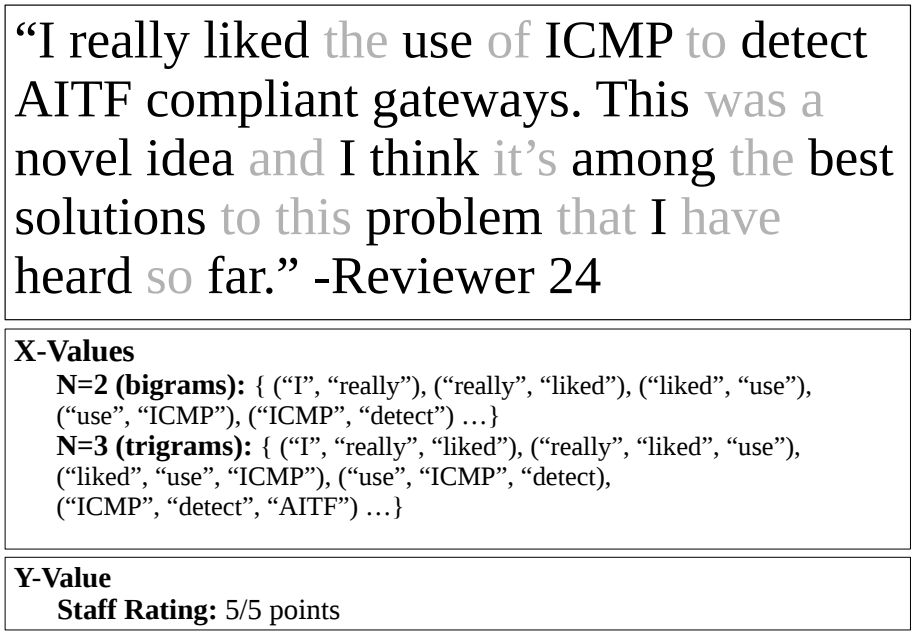


Figure 3: Example construction of features for attempted machine-learning methodology (*Computer Network Security*)

4 Analysis

Returning to our initial questions (Section 1), we wish to quantify students' security-focus by examining the contents of the reviews they have produced in past sections of the courses under study (Section 3). We also wish to determine the degree to which they do so in an abstract way or a concrete way. Finally, we will explore how the different configuration parameters between the two courses may account for differences in the data. In order to observe these differences, we required a qualitative methodology for classifying statements that reviewers make when discussing their peers' work.

4.1 Machine-Learning Trials

In the formative stages of this project, we considered the use of some machine-learning techniques and textual analysis tools to automatically analyze the corpus of review text from *Computer Network Security*. The reviews from *Software Security Engineering*, by contrast,

we consider too short to create meaningful models, and they lack a ground-truth with which to build such models. We attempted to use a Python natural-language toolkit (NLTK) to construct a linear model of the *Computer Network Security* review contents.

To process the text for model creation, we first filtered out *stop words*, or words that "help build ideas but do not carry any significance themselves"[24]. Then we constructed a sets of *N-grams* (sets of N adjacent words) appearing in a particular review. We constructed these sets for values of N between 1 and 5. Using these sets of N-grams and their frequencies as features of our data, and using the scores given to the reviews by the course staff as a ground truth, we used NLTK to produce a linear model relating the N-gram contents to the review score (Figure 3).

Using one half of the reviews to train the model and the other half to test it, we found that our model performed very poorly (with an accuracy indistinguishable from random selection). While we were initially skeptical of linear model's ability to accurately predict scores based only on a simple regression, after examining the generated model, we found that the review scores do not exhibit enough variance to reliably train any model, and we were dissuaded from pursuing further machine-learning techniques using the review scores as ground truth.

Having realized the lack of ground truth, we developed a qualitative analysis methodology based on manual coding of review data. We then used quantitative analysis techniques to analyze the features we identified in the review data. This methodology is similar to the techniques employed in Politz *et al.*'s work on peer review of tests [7] and is further supported by Basit's 2003 paper on manual and electronic coding in qualitative analysis [25].

4.2 Coding Rubric

Our manual coding rubric (Appendix B) considers each statement (defined at the sub-sentence level, i.e. a clause expressing a single suggestion or statement of fact) in a review and classifies it in *only* one of three *categories*:

"Overall a good quality paper with nice visuals and specific, but not overbearing, details. The only thing I would recommend is to take a look at vulnerabilities in security in this type of system, even if those security issues will never be addressed directly. It it good to have an explicit outline of the systems strengths and weaknesses." -Reviewer 63

- "good quality": general praise
- "nice visuals and specific, but not overbearing, details": neutral comment (no suggestion, only a statement) on document structure
- "take a look at vulnerabilities in security in this type of system": abstract (no direct instruction or suggestion) security-related comment
- "have an explicit outline of the systems strengths and weaknesses": concrete (directly asks the author to develop an outline) suggestion about the structure of the document

Figure 4: Example review and identified codes (*Computer Network Security*)

- **technical**: implementation details or design implications (except statements that fit the security-related category below)
- **security-related**: a technical comment which is intrinsically security concerned e.g. use of nonce values, using strong hashing algorithms
- **structural**: information presentation e.g. document layout

Further, it classifies each comment as belonging to one of the following *moods*:

- **neutral**: a matter-of-fact declaration or statement
- **abstract**: non-specific suggestions, e.g. "this implementation is not secure"
- **concrete**: specific and actionable suggestions, e.g. "you use MD5, which is not secure. . . , use SHA-2 instead"

We also code for statements which express one of the following general sentiments:

- **general praise**: statements which praise the work generally, e.g. "this paper is good"
- **confusion**: the reviewer says "I do not understand. . . ", or something to that effect
- **learning**: the reviewer expresses their own learning, e.g. "[your idea] is a unique case that I hadn't thought of before"
- **direct praise/criticism**: statements which praise or criticize a specific component of the work
- **rudeness**: direct, personal insults or any other inappropriate discourse

Each review (in both courses) is composed of four independent sections. Using the rubric above, we coded each section of each review individually. We allowed each classification to apply more than once to a particular section, so that we could understand the degree to which particular sections were focused on a particular class of commentary and the relative frequency of each kind of commentary (Figure 4). We also noted cases in which reviews contained no text as well as cases in which reviews contained words, but no real meaning (in the context of this rubric).

4.2.1 Sampling

We chose a random sample of thirty reviews from the first assignment in both courses (out of 144 reviews in *Software Security Engineering* and 90 reviews in *Computer Network Security*) and coded these reviews. Reviews that were statistical outliers in terms of review lengths (in words) were excluded from the selection pool. We then chose a sample of eight reviews from the original thirty (for both courses) which we felt exhibited the most distinctive traits (such as high amounts of technical/structural feedback, review length, etc.) and coded a review by their authors for each subsequent assignment.

5 Results

We coded 100 reviews in total (30 reviews of the first assignment per course, plus the additional 24 reviews from *Software Security Engineering* and an additional 16 reviews from *Computer Network Security*). In this section, we describe some of the observed characteristic differences between the reviews in each of the two courses that we gathered from.

5.1 Survey Results

We opened our survey (Section 3.2) for responses to all students in an active section of *Computer Network Security* (about 65 total) and received responses from 17. As our response rate is somewhat low, we used the survey results (Appendix C) only to guide further exploration of the review contents, and we discuss survey results only when they suggest clarifications to our findings in the review contents.

5.1.1 Survey Respondents

Of our 17 survey respondents, 4 (23.5%) were juniors and 13 (76.5%) were seniors. When we asked our respondents whether or not they had used peer review before,

- 16 reported that they had used peer review in **another course**
- 6 reported that they had participated in an **industrial code review**
- 2 reported that they had encountered peer review in an **academic** setting (for a publication in an academic venue)

5.2 Comparison of first assignments

The first component of our analysis focuses on differences observed between the reviews performed on the first submission in each course. We examine the tendency for students

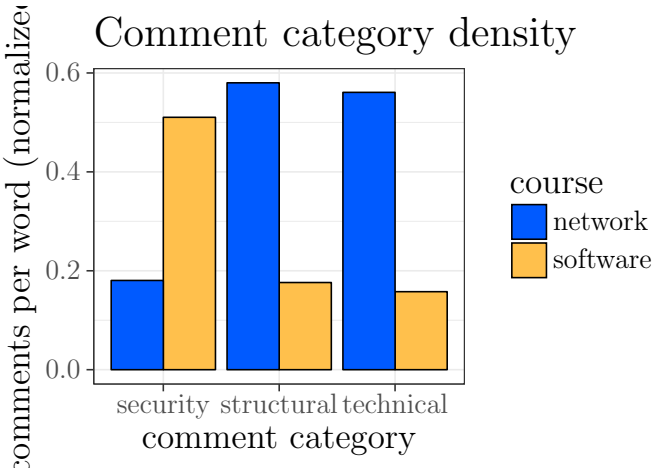


Figure 5: Relative densities of each category of comment, by course

in each course to offer commentary of each category and mood described by our rubric (Section 4.2).

5.2.1 Security-orientation

We compare both the absolute quantities of codes in the entire corpus of reviews for each course and their relative densities. When we examine the absolute quantities, we observe that *Software Security Engineering* and *Computer Network Security* both share roughly the same amount of security commentary (in total, we coded 56 instances of the security category in *Computer Network Security* and 55 in *Software Security Engineering*), while the *Computer Network Security* reviews have much more structural and technical commentary. However, when we examine the density (relative frequency of occurrence) of security codes, normalized for review length (Figure 5), we find that the *Software Security Engineering* reviews are far more *security-oriented* than the others.

In other words, while students write less overall content in *Software Security Engineering*, a larger portion of it is dedicated to security-oriented commentary compared to the *Computer Network Security* students, who generate much more content related to the structure of the artifact and other technical details. Our survey respondents generally indicated that structural feedback was easier to generate (one respondent remarked that the concepts for

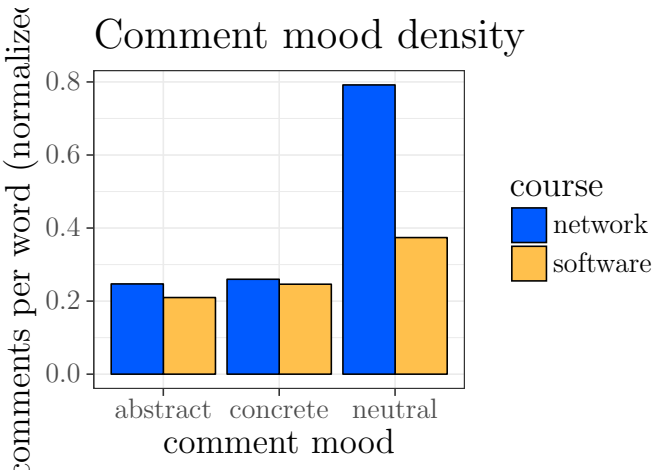


Figure 6: Relative densities of each comment mood, by course

document structure analysis are "taught in middle or high school").

5.2.2 Reviewer Moods

We found that students in both *Software Security Engineering* and *Computer Network Security* use the abstract and concrete moods at roughly the same frequency. The notable difference, when comparing the mood of the reviews, is that students in *Computer Network Security* use the neutral mood at roughly twice the frequency of *Software Security Engineering* students (Figure 6). However, more than half of these neutral statements (51.2%) are located in the "Summary" section of the *Computer Network Security* reviews. This section asks reviewers to simply describe the artifact under review, so a great deal of neutral commentary is to be expected. If we adjust our computation to discard the "summary" section, then the adjusted densities are almost identical (within 2 words per incidence of a neutral comment).

5.2.3 Security Commentary Mood

At the outset of this project, we asked whether students would comment on security from a mostly concrete or abstract perspective. We found that students split their review contents close to evenly between the two moods, with abstract commentary being very slightly more common. Notably, the two courses exhibit the same distribution of moods of security

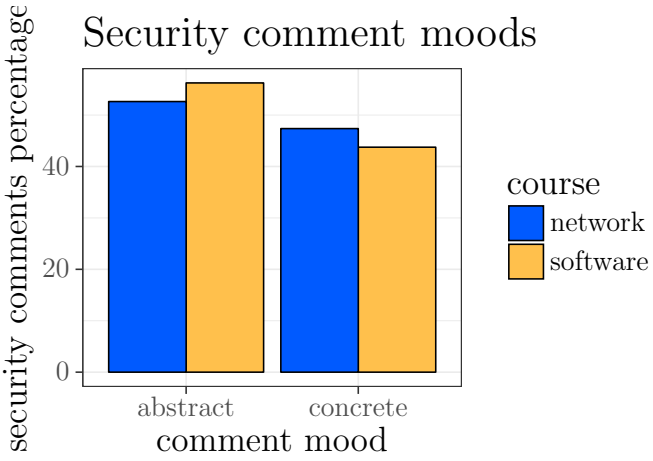


Figure 7: Concrete security feedback *vs.* abstract security feedback

commentary (Figure 7).

5.3 Development over multiple assignments

We now address the development of review contents over time (across multiple assignments). While we observe an interesting trend in the lengths of reviews overall, we do not find any indication in the sample we analyzed that reviews by the same reviewer share any meaningful characteristics. Broadly, we find that those reviewers who wrote the longest reviews in one assignment were among the longest reviewers in subsequent assignments, but even so there exists a significant dispersion of review lengths even among individual reviewers.

5.3.1 Review Lengths

Notably, the lengths of reviews in both courses decrease over time. More notably, they seem to decrease at about the same rate. The average length of a review of the first assignment in *Computer Network Security* was 212.57 words, and for the third (final) assignment the average length was 126.47 words. In *Software Security Engineering*, the average length of a review was 59.7 words for the first assignment and 35.57 words for the final. In both courses, the decrease in length between the first and final assignments is 40%.

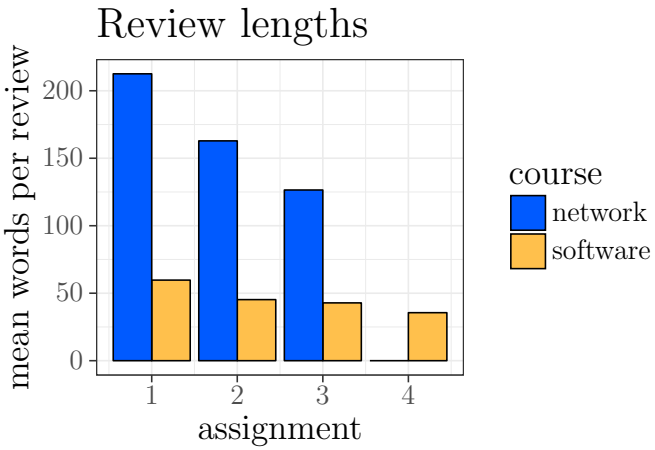


Figure 8: Average review lengths, by course (*Computer Network Security* did not have a fourth assignment)

6 Discussion

Instructors who choose to use peer review in cybersecurity courses face a series of decisions regarding how precisely to implement peer reviews for maximum effectiveness. It is important to note that our work is primarily exploratory. Our goal was to describe the characteristics of peer reviews performed in the context of cybersecurity education. We do not describe any statistically significant correlation, and educators who are interested in instrumenting peer review for cybersecurity education should keep this in mind. We suggest some applications and interpretations of the findings in this paper.

We are primarily stricken by the inversion of priorities between *Software Security Engineering* and *Computer Network Security* (Figure 5). We speculate that this is likely due to the types of artifacts that the student reviewers were presented for analysis. The highly schematic nature of the exploit lists that *Software Security Engineering* students prepared offered little structure to comment on. On the other hand, the free-written documents under study in *Computer Network Security* provided ample opportunity to comment on structure. If the goal of the peer review system is to engage students with the topic of cybersecurity, then both courses appear to prompt (on average) the same amount of security-specific comments. We also find it noteworthy that, despite the differences in peer review environment, artifacts,

and prompts, the students deliver roughly the same distribution of abstract *vs.* concrete commentary in both courses. This may imply that the distribution of these moods is intrinsic to the review process.

We encourage readers not to dwell too heavily on the decrease in review length over time (Figure 8). This decay may indicate that students become fatigued of the review process, but it may also indicate that reviewers become more efficient at delivering effective reviews. *Computer Network Security* students who responded to our survey suggested that reviews improved over time (94.1% have a neutral or more-positive level of agreement with the statements "over the course of the term, I became better able to provide useful feedback", and "over the course of the term, the feedback I received from my peers became more useful"), but they lamented in the comment section of the survey that they did not have enough time to prepare the reviews or respond to them. They did, however, agree that "peer review is a useful skill" (94.1% *agree* or *strongly agree*) and that "peer review has made worthwhile use of course time" (70.6% *agree* or *strongly agree*).

A variety of factors may explain the increased length of the peer reviews in *Computer Network Security* (Figure 8). We suspect the most likely contributing factors to this difference are the instructor's expectations for peer review (some instructors may desire a more formal process with longer reviews). However, the students may also have felt compelled to write longer reviews due to the assessment of their reviews as a component of their course grade. Educators interested in implementing peer review in cybersecurity courses (and possibly other specific domains) should note that longer reviews do not necessarily indicate that students are engaging with the course topic more than students who write short reviews.

Finally, reflecting on our early question about whether or not students have consistency across reviews of sequential assignments, we suspect two things: (1) our sampling methodology may have been flawed and (2) reviewers' commentary may be more of a reflection of the artifact they review as opposed to an internal "voice." By picking examples for anlysis over the course of multiple assignments by virtue of their relative extremity, we may have

inadvertently chosen a segment of the reviews which was the most likely to exhibit future change. Furthermore, as reviews can be seen as responses to the authors' work, it seems likely that reviewers' decisions in constructing those reviews may be influenced by the characteristics of the author's original work more than by a reviewer's own writing style.

7 Future Work

While our survey showed that our narrow sample of *Computer Network Security* students slightly preferred technical and security-oriented reviews to those that commented on structural components, we are interested in (1) how consumption of reviews may influence future reviews given to other students and (2) which types of reviews are more likely to be incorporated into the authors' future works.

Our analysis focuses on *what* we observe in peer reviews expressed in two different contexts. Future research might begin to investigate *why* these differences manifest. We are interested in what motivates students to comment on cybersecurity as opposed to other topics. Is it the instructor? The course? Prior experience? Future studies should collect more information about students' prior exposure to peer review and examine how this may affect their experience. We also are interested in how peer review contributes to learning outcomes as well as *what* it contributes. While the literature on this topic strongly suggests that peer review practice in the cybersecurity context may reinforce students' knowledge, we would like to observe and quantify this effect so that it can be more precisely articulated.

Our initial foray into machine learning techniques was unsuccessful, but we are still interested in potential applications of machine learning to understanding peer review. Given the coding methodology described in this paper, perhaps a successful machine-learning algorithm could learn to classify peer reviews according to our rubric. Atapattu and Falkner experimented successfully with the use of machine-learning techniques to automatically classify forum topics in MOOC environments according to their contents [26]. They hope

to use this technique to empower MOOC authors to analyze the progression of their course through the concerns, comments, and posts of their students. Similarly, an automatic classifier might be used to gain insights into the attitudes of students approaching peer reviews.

Broadly, we encourage more data collection under more controlled circumstances. While the high variation in process between *Computer Network Security* and *Software Security Engineering* was useful to show the differences and similarities between the results of the courses at a high-level, any analysis which seeks to establish significant correlations will need data collected under more controlled circumstances. Since the reviews in our courses are assigned and collected using an online system, it may be a prime candidate for A/B testing.

References

[1] K. Topping, "Peer Assessment between Students in Colleges and Universities," *Review of Educational Research*, vol. 68, no. 3, p. 249, 1998.

[2] K. Lundstrom and W. Baker, "To give is better than to receive: The benefits of peer review to the reviewer's own writing," *Journal of Second Language Writing*, vol. 18, pp. 30–43, Mar. 2009.

[3] F. Dochy, M. Segers, and D. Sluijsmans, "The use of self-, peer and co-assessment in higher education: A review," *Studies in Higher Education*, vol. 24, pp. 331–350, Jan. 1999.

[4] E. F. Gehringer, "Electronic Peer Review and Peer Grading in Computer-science Courses," in *Proceedings of the Thirty-second SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '01, (New York, NY, USA), pp. 139–143, ACM, 2001.

[5] J. G. Politz, S. Krishnamurthi, and K. Fisler, "CaptainTeach: A Platform for In-flow Peer Review of Programming Assignments," in *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, (New York, NY, USA), pp. 332–332, ACM, 2014.

[6] D. Clarke, T. Clear, K. Fisler, M. Hauswirth, S. Krishnamurthi, J. G. Politz, V. Tirronen, and T. Wrigstad, "In-Flow Peer Review," in *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, ITiCSE-WGR '14, (New York, NY, USA), pp. 59–79, ACM, 2014.

[7] J. G. Politz, S. Krishnamurthi, and K. Fisler, "In-flow Peer-review of Tests in Test-first Programming," in *Proceedings of the Tenth Annual Conference on International Computing Education Research*, ICER '14, (New York, NY, USA), pp. 11–18, ACM, 2014.

[8] J. Cohen, *Best Kept Secrets of Peer Code Review*. Smart Bear, Inc., 2006.

[9] M. M. Nelson and C. D. Schunn, "The nature of feedback: how different types of peer feedback affect writing performance," *Instructional Science*, vol. 37, pp. 375–401, July 2009.

[10] E. F. Gehringer, L. M. Ehresman, S. G. Conger, and P. A. Wagle, "Work in Progress: Reusable Learning Objects Through Peer Review: The Expertiza Approach," in *Proceedings. Frontiers in Education. 36th Annual Conference*, pp. 1–2, Oct. 2006.

[11] C. M. Hicks, C. A. Fraser, P. Desai, and S. Klemmer, "Do Numeric Ratings Impact Peer Reviewers?," in *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, L@S '15, (New York, NY, USA), pp. 359–362, ACM, 2015.

[12] "Fact sheet: Cybersecurity National Action Plan." `https://www.whitehouse.gov/the-press-office/2016/02/09/fact-sheet-cybersecurity-national-action-plan`. Online; accessed November 10, 2016.

[13] M. Timchenko and D. Starobinski, "A Simple Laboratory Environment for Real-World Offensive Security Education," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '15, (New York, NY, USA), pp. 657–662, ACM, 2015.

[14] P. J. Wagner and J. M. Wudi, "Designing and Implementing a Cyberwar Laboratory Exercise for a Computer Security Course," in *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '04, (New York, NY, USA), pp. 402–406, ACM, 2004.

[15] C. E. Kaucher and J. H. Saunders, "Building an information assurance laboratory for graduate-level education," in *6th National Colloquium for Information System Security Education, Redmond, WA*, 2002.

[16] R. Shumba, "Towards a More Effective Way of Teaching a Cybersecurity Basics Course," in *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education*, ITiCSE-WGR '04, (New York, NY, USA), pp. 108–111, ACM, 2004.

[17] S. Cooper, C. Nickell, L. C. Pérez, B. Oldfield, J. Brynielsson, A. G. Gökce, E. K. Hawthorne, K. J. Klee, A. Lawrence, and S. Wetzel, "Towards Information Assurance (IA) Curricular Guidelines," in *Proceedings of the 2010 ITiCSE Working Group Reports*, ITiCSE-WGR '10, (New York, NY, USA), pp. 49–64, ACM, 2010.

[18] D. Manson and R. Pike, "The Case for Depth in Cybersecurity Education," *ACM Inroads*, vol. 5, pp. 47–52, Mar. 2014.

[19] R. B. Vaughn, D. A. Dampier, and M. B. Warkentin, "Building an Information Security Education Program," in *Proceedings of the 1st Annual Conference on Information Security Curriculum Development*, InfoSecCD '04, (New York, NY, USA), pp. 41–45, ACM, 2004.

[20] MITRE Corporation, The, "Secure Code Review," 2011.

[21] A. Bacchelli and C. Bird, "Expectations, Outcomes, and Challenges of Modern Code Review," in *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, (Piscataway, NJ, USA), pp. 712–721, IEEE Press, 2013.

[22] "Captain Teach." `https://www.captain-teach.org/`. Online; accessed October 30, 2016.

[23] K. Argyraki and D. R. Cheriton, "Scalable Network-layer Defense Against Internet Bandwidth-flooding Attacks," *IEEE/ACM Trans. Netw.*, vol. 17, pp. 1284–1297, Aug. 2009.

[24] A. Rajaraman and J. D. Ullman, "Data Mining," in *Mining of Massive Datasets:*, pp. 1–17, Cambridge: Cambridge University Press, Oct. 2011. DOI: 10.1017/CBO9781139058452.002.

[25] T. Basit, "Manual or electronic? The role of coding in qualitative data analysis," *Educational Research*, vol. 45, pp. 143–154, June 2003.

[26] T. Atapattu and K. Falkner, "A Framework for Topic Generation and Labeling from MOOC Discussions," in *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, L@S '16, (New York, NY, USA), pp. 201–204, ACM, 2016.

A CS4404 Student Survey

The following pages show a PDF rendering of the survey (produced and administered using Google Forms) that we distributed electronically to students participating in *Computer Network Security* in A-Term of 2016.

Peer Review Survey

This survey will ask several questions about your thoughts on the Peer Review process that you participated in as a student in CS4404. The survey is anonymous, so please answer the questions honestly, as your answers will inform future modifications to the process.

At the end of the survey, you will be asked to enter your email address if you wish to enter a raffle for an Amazon gift-card.

* Required

1. **In which of the following contexts have you participated in a Peer Review system previously (check all that apply)? ***
   Check all that apply.

   - [ ] Other courses
   - [ ] Professional Code Review
   - [ ] Academic Review (Journal Publications)
   - [ ] Other: _____

2. **Select your Class Standing ***
   Mark only one oval.

   - ◯ Freshman
   - ◯ Sophomore
   - ◯ Junior
   - ◯ Senior
   - ◯ Graduate Student
   - ◯ Other

Peer Review Experience

In this section, we will ask several questions about your peer-review experience with possible answers on a scale from 1 to 5.

Answer the following question on a scale of "Never" to "Very Frequently."

3. **How frequently did you incorporate suggestions from your peers' reviews of your preliminary submissions into your final submissions?**
   Mark only one oval.

	1	2	3	4	5	
Never	◯	◯	◯	◯	◯	Very Frequently

Answer the following questions on a scale of "Strongly Disagree" to "Strongly Agree."

4. **I provided useful feedback on my peers' submissions.**
   Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

5. **My peers provided useful feedback to me on my submissions.**
   Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

6. **Over the course of the term, I became better able to provide useful feedback.**
   Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

7. **Over the course of the term, the feedback I received from my peers became more useful.**
   Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

8. **I easily identified strengths to comment on in my peers' work.**
   Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

9. **I easily identified weaknesses to comment on in my peers' work.**
   Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

10. **The instructor's expectations for review content were clearly defined and understandable.**
    Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

Computer Security and Peer Review

This section will ask questions about Peer Review as it relates to the study of Computer Security.

Answer the following questions on a scale of "Never" to "Very Frequently."

11. **How frequently did you identify security vulnerabilities in others' designs during your review of their work?**
    Mark only one oval.

	1	2	3	4	5	
Never	◯	◯	◯	◯	◯	Very Frequently

12. **How frequently did you identify security vulnerabilities in your own designs as a result of peers' reviews on your own work?**
    Mark only one oval.

	1	2	3	4	5	
Never	◯	◯	◯	◯	◯	Very Frequently

13. **How frequently did you identify security vulnerabilities in your own designs as a result of reviewing another student's work?**
Mark only one oval.

	1	2	3	4	5	
Never	◯	◯	◯	◯	◯	Very Frequently

Technical Feedback vs. Structural Feedback

The following questions ask you to compare and contrast reviews which focused mostly on technical details with those that focused on the structure of the files under review. Technical details would include, for example, the content and layout of packet headers, correct use of "nonce" values, identification of specific vulnerabilities, and any other details which involve a technical understanding of the implementation of the protocol which you studied in CS4404.

Structural feedback refers to any comments or criticisms relating to the way the information was conveyed in the documents under review. For example, structural details would include the use (or misuse) of graphs, paragraph structure, logical flow, clarity of writing, etc. for the sake of conveying information to the reader.

Answer the following questions on a scale of "Mostly Technical" to "Mostly Structural," given the definitions of those terms outlined above.

14. **The reviews I received on my work were:**
Mark only one oval.

	1	2	3	4	5	
Mostly Technical	◯	◯	◯	◯	◯	Mostly Structural

15. **The reviews I gave to other students were:**
Mark only one oval.

	1	2	3	4	5	
Mostly Technical	◯	◯	◯	◯	◯	Mostly Structural

16. **The most useful reviews are:**
Mark only one oval.

	1	2	3	4	5	
Mostly Technical	◯	◯	◯	◯	◯	Mostly Structural

17. **Which of technical or structural feedback did you find easier to identify and comment on? Why?**

Software Security & Network Security Comparison

18. **Have you taken CS4401 - Software Security Design & Analysis at WPI?** *
Mark only one oval.

- ◯ Yes, and I used Captain Teach for Peer Review in CS4401.
- ◯ Yes, but we did not use a peer-review system, or we used a system other than Captain Teach. *Skip to question 23.*
- ◯ No *Skip to question 23.*

Comparison to Software Security

You indicated that you have participated in a section of CS4401 "Software Security" which used Captain Teach. This section contains some questions which ask you to compare and contrast that experience with the experience in this class.

19. **Did you prefer the more specific questions (e.g. "Do you think these exploits are qualitatively different from each other?") that were asked on reviews in Software Security to the free-response categories in Network Security or vice-versa? Why?**

20. **Which class provided more actionable feedback? Why? For example, did one class provide more technical details vs. structural details? Did one class provide more content or more heavily condensed content? Was the subject matter in one course more approachable?**

21. **Indicate which course's peer review style you preferred, in general.**
Mark only one oval.

	1	2	3	4	5	
Software Security	◯	◯	◯	◯	◯	Network Security

22. **What aspects of each system lead you to your previous answer?**

Peer Review (in General)

In this section, you have the opportunity to provide any additional thoughts on the Peer Review process that were not covered by the previous sections. Responses to these questions are optional.

23. **Peer Review is a useful skill.**
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

24. **Peer Review has made worthwhile use of course time.**
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

25. **Do you have any additional comments on what you liked/dislikes about peer review, or about how we can improve the process in the future?**

Raffle Entry

Provide your email address in the form below if you wish to be entered into the raffle for an Amazon gift card. The anonymity of your answers is still guaranteed by the survey's authors if you provide your email address. Your name will not be shared with the course staff.

26. **Please enter your WPI email address below:**

B Classification Rubric

Each comment is categorized as either (1) matching one of the general classifiers below or (2) matching a combination of exactly one mood and one category code

Categories	**Criteria**
Structural Comment	• comments about the use of figures, tables, or graphs • comments about clarity of writing or ambiguity in descriptions • ex. "The figure demonstrating the RR shim is very clear."
Technical Comment	• comments about specific implementation details • comments about implications of design decisions • ex. "The size of your RR shim will add additional overhead to your system."
Security Comment	• commentary which meets the above criteria for a *Technical Comment* but is also security-related • commentary relating to a system vulnerability • ex. "How will your router determine that a forwarding request is authentic?"

Moods	**Criteria**
Concrete Suggestion	• Criticisms that come "bundled" with a solution or concrete instruction • ex. "Change the RR header to incorporate a nonce value to prevent forgery"
Abstract Suggestion	• Criticisms that do not present immediate solutions as part of the feedback • ex. "Your paper is is too complicated."
Neutral Comment	• Comments which state a fact, but which do not offer suggestions or criticisms • ex. "The first exploit allows the attacker to XYZ, and the second allows IJK."

General Classifiers	**Criteria**
Positive Feedback	ex. "This paper is well done!"
Confusion	ex. "I am not sure what you mean by XYZ."
Learning	ex. "I am going to incorporate XYZ into my zolution."
Direct Praise	ex. "Good job with the descriptions of the router functionality."
Direct Criticism	ex. "Your use of the netfilter library was a poor choice."
Rudeness	direct insults or other inappropriate discourse
No Comment	the comment contains words, but is devoid of meaning or suggestion
No Response	the comment is blank

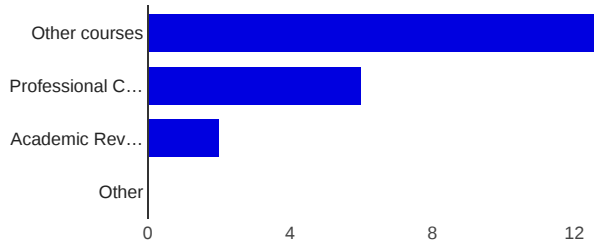
C CS4404 Student Survey Results

The following pages show a PDF rendering of the survey results (produced by Google Forms). These results do not include the free-response questions, which may contain identifying information.

17 responses

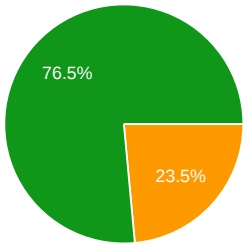
Summary

In which of the following contexts have you participated in a Peer Review system previously (check all that apply)?



Other courses	**16**	94.1%
Professional Code Review	**6**	35.3%
Academic Review (Journal Publications)	**2**	11.8%
Other	**0**	0%

Select your Class Standing

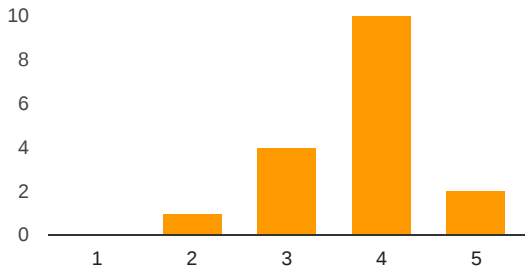


Freshman	**0**	0%
Sophomore	**0**	0%
Junior	**4**	23.5%
Senior	**13**	76.5%
Graduate Student	**0**	0%
Other	**0**	0%

Peer Review Experience

Answer the following question on a scale of "Never" to "V ery Frequently."

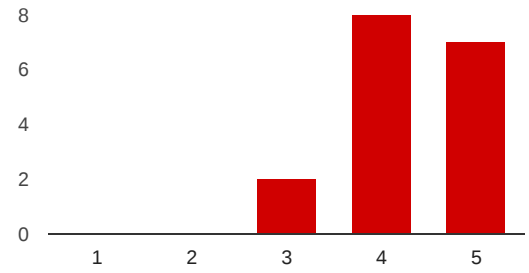
How frequently did you incorporate suggestions from your peers' reviews of your preliminary submissions into your final submissions?



Never:	1	**0**	0%
	2	**1**	5.9%
	3	**4**	23.5%
	4	**10**	58.8%
Very Frequently:	5	**2**	11.8%

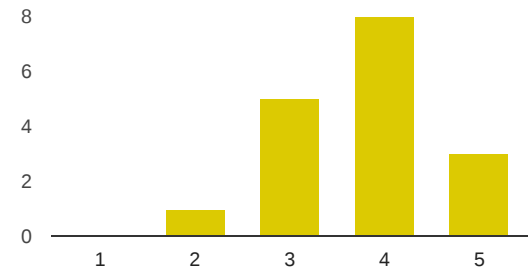
Answer the following questions on a scale of "Strongly Disagree" to "Strongly Agree."

I provided useful feedback on my peers' submissions.



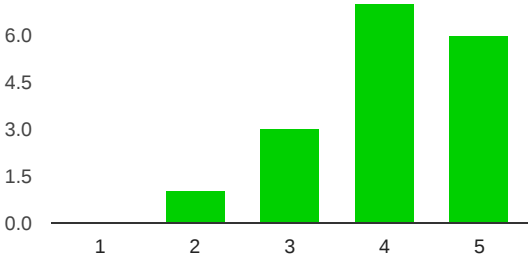
Strongly Disagree:	1	**0**	0%
	2	**0**	0%
	3	**2**	11.8%
	4	**8**	47.1%
Strongly Agree:	5	**7**	41.2%

My peers provided useful feedback to me on my submissions.



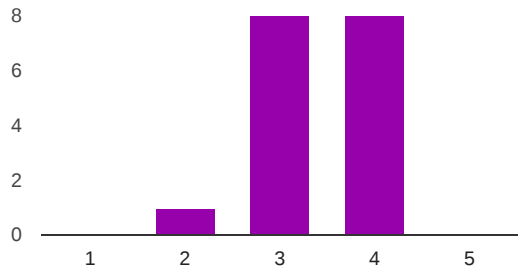
Strongly Disagree:	1	**0**	0%
	2	**1**	5.9%
	3	**5**	29.4%
	4	**8**	47.1%
Strongly Agree:	5	**3**	17.6%

Over the course of the term, I became better able to provide useful feedback.



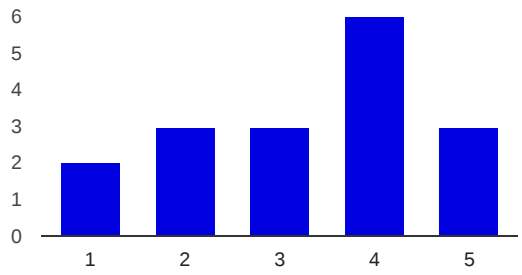
Strongly Disagree:	1	**0**	0%
	2	**1**	5.9%
	3	**3**	17.6%
	4	**7**	41.2%
Strongly Agree:	5	**6**	35.3%

Over the course of the term, the feedback I received from my peers became more useful.



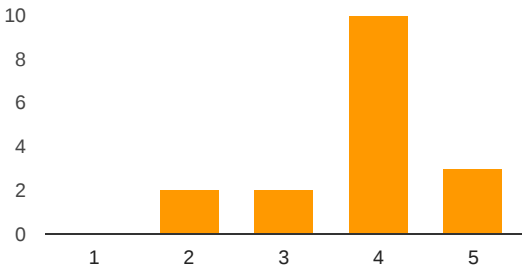
4	**8**	47.1%
Strongly Agree: 5	**0**	0%

I easily identified strengths to comment on in my peers' work.



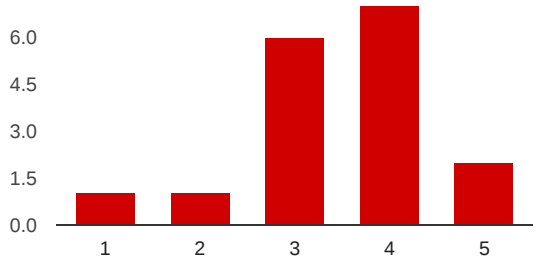
Strongly Disagree: 1	**2**	11.8%
2	**3**	17.6%
3	**3**	17.6%
4	**6**	35.3%
Strongly Agree: 5	**3**	17.6%

I easily identified weaknesses to comment on in my peers' work.



Strongly Disagree:	1	**0**	0%
	2	**2**	11.8%
	3	**2**	11.8%
	4	**10**	58.8%
Strongly Agree:	5	**3**	17.6%

The instructor's expectations for review content were clearly defined and understandable.

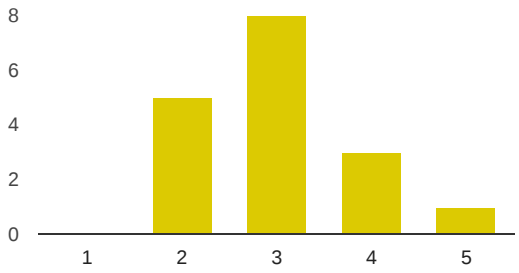


Strongly Disagree:	1	**1**	5.9%
	2	**1**	5.9%
	3	**6**	35.3%
	4	**7**	41.2%
Strongly Agree:	5	**2**	11.8%

Computer Security and Peer Review

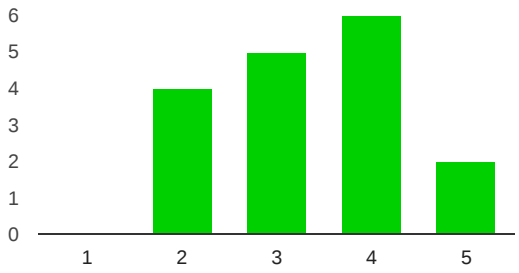
Answer the following questions on a scale of "Never" to "Very Frequently."

How frequently did you identify security vulnerabilities in others' designs during your review of their work?



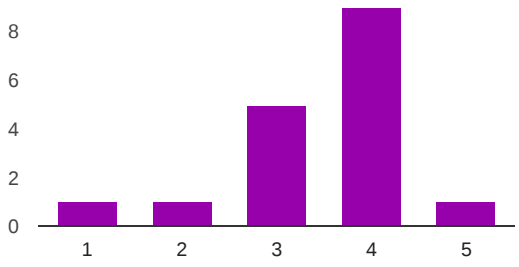
Never: 1	**0**	0%
2	**5**	29.4%
3	**8**	47.1%
4	**3**	17.6%
Very Frequently: 5	**1**	5.9%

How frequently did you identify security vulnerabilities in your own designs as a result of peers' reviews on your own work?



Never: 1	**0**	0%
2	**4**	23.5%
3	**5**	29.4%
4	**6**	35.3%
Very Frequently: 5	**2**	11.8%

How frequently did you identify security vulnerabilities in your own designs as a result of reviewing another student's work?

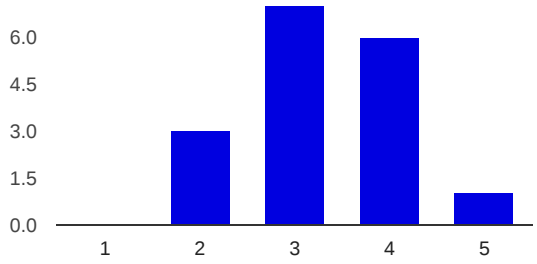


Never: 1	**1**	5.9%
2	**1**	5.9%
3	**5**	29.4%
4	**9**	52.9%

Very Frequently: 5	**1**	5.9%

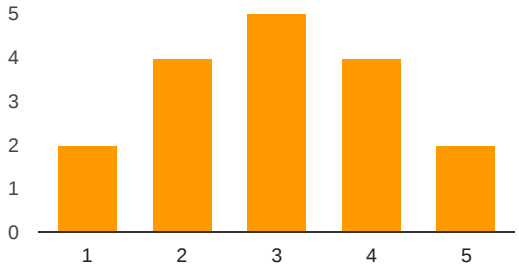
Technical Feedback vs. Structural Feedback

The reviews I received on my work were:



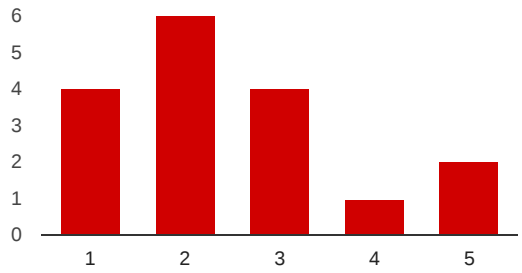
Mostly Technical: 1	**0**	0%
2	**3**	17.6%
3	**7**	41.2%
4	**6**	35.3%
Mostly Structural: 5	**1**	5.9%

The reviews I gave to other students were:



Mostly Technical: 1	**2**	11.8%
2	**4**	23.5%
3	**5**	29.4%
4	**4**	23.5%
Mostly Structural: 5	**2**	11.8%

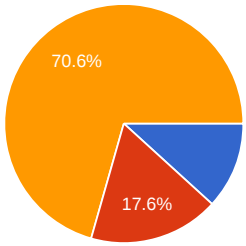
The most useful reviews are:



Mostly Technical:	1	**4**	23.5%
	2	**6**	35.3%
	3	**4**	23.5%
	4	**1**	5.9%
Mostly Structural:	5	**2**	11.8%

Software Security & Network Security Comparison

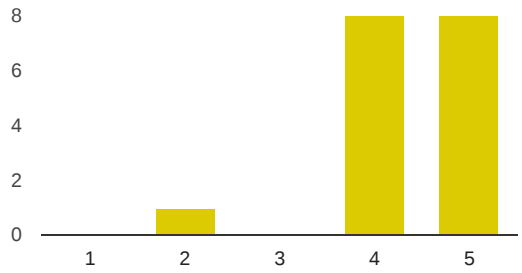
Have you taken CS4401 - Software Security Design & Analysis at WPI?



Yes, and I used Captain Teach for Peer Review in CS4401.	**2**	11.8%
Yes, but we did not use a peer-review system, or we used a system other than Captain Teach.	**3**	17.6%
No	**12**	70.6%

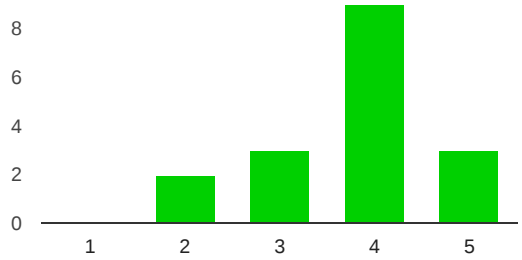
Peer Review (in General)

Peer Review is a useful skill.



Strongly Agree: 5 **8** 47.1%

Peer Review has made worthwhile use of course time.



Strongly Disagree: 1	**0**	0%
2	**2**	11.8%
3	**3**	17.6%
4	**9**	52.9%
Strongly Agree: 5	**3**	17.6%