Check for updates

ACM DIGITAL LIBRARY | Association for Computing Machinery | acm open

Latest updates: https://dl.acm.org/doi/10.1145/3769994.3770022

PDF Download
3769994.3770022.pdf
14 March 2026
Total Citations: 0
Total Downloads: 287

Published: 10 November 2025

Citation in BibTeX format

Koli Calling '25: 25th Koli Calling International Conference on Computing Education Research
November 11 - 16, 2025
Koli, Finland

RESEARCH-ARTICLE

Observations of Highly Successful, Distributed Teams in a K-12 Online Security Competition

EMMA HOGAN, Bard College, Red Hook, NY, United States

ALEJANDRO CUEVAS VILLALBA, Carnegie Mellon University, Pittsburgh, PA, United States

HANAN HIBSHI, Carnegie Mellon University, Pittsburgh, PA, United States

NICOLAS CHRISTIN, Carnegie Mellon University, Pittsburgh, PA, United States

Open Access Support provided by:

Carnegie Mellon University

Bard College

Koli Calling '25: Proceedings of the 25th Koli Calling International Conference on Computing Education Research (November 2025)
https://doi.org/10.1145/3769994.3770022
ISBN: 9798400715990

Observations of Highly Successful, Distributed Teams in a K-12 Online Security Competition

Emma Hogan
Bard College
Annandale-on-Hudson, NY, USA
ehogan@bard.edu

Alejandro Cuevas Villalba
Carnegie Mellon University
Pittsburgh, PA, USA
acuevasv@andrew.cmu.edu

Hanan Hibshi
Carnegie Mellon University
Pittsburgh, PA, USA
King Abdulaziz University
Jeddah, Saudi Arabia
hhibshi@cmu.edu

Nicolas Christin
Carnegie Mellon University
Pittsburgh, PA, USA
nicolasc@andrew.cmu.edu

Abstract

Security competitions in computing education model real-world crowd-sourced security (e.g., bug bounty programs), which has grown dramatically in recent years and has become the main source of software security reviews for many companies. This study explores practices and dynamics of high-performing teams of high school computing students in a K-12 computer security competition. While existing literature covers dynamics of security teams in structured organizations and student team formation for software engineering projects, additional insights are to be gained from this setting where students in this study a) self-selected into remote or hybrid teams, and b) developed collaborative strategies that enabled them to outperform thousands of teams on computer security tasks. We aim to fill this gap by addressing the following research questions regarding high-performing teams of high school computing students collaborating on computer security tasks:

(1) *How, why, and through which assembly mechanisms do teams form?*
(2) *What technological needs do teams have, and how do they address them?*
(3) *How do teams scope and distribute tasks, roles, and responsibilities?*
(4) *What motivations and limitations do these teams have?*
(5) *What factors and strategies seem to contribute to teams' performance?*

We conducted focus groups with the top five teams (out of 18,201 participating teams) of a computer security Capture-the-Flag (CTF) competition, all consisting of high school students in the U.S. To analyze the qualitative data collected using a structured-interview format, we analyzed data using grounded theory to address the research questions.We find that the teams in our study adopted a set of strategies centered on specialties, which allowed them to reduce issues relating to dispersion, double work, and lack of previous collaboration. We also connect these results to prior frameworks to conceptualize crowd work, including how students organically adopted existing strategies for task decomposition, hierarchical/reputational organization, task assignment, and collaboration. This work has implications in both computer security education and distributed teamwork strategies developed among high school students pre-COVID19 pandemic, which is before remote work became more common among this age group. The teams in our study model how scaling and distributing security work among team members is feasible, productive and beneficial. In addition, we identify various areas for future work, such as issues of social identity in high-skilled crowd work environments.

CCS Concepts

• **Social and professional topics** → **Computing education**; **K-12 education**; **Computer science education**; • **Security and privacy**;

Keywords

Computer Security Education, K-12, CTF, Distributed Teams

ACM Reference Format:
Emma Hogan, Alejandro Cuevas Villalba, Hanan Hibshi, and Nicolas Christin. 2025. Observations of Highly Successful, Distributed Teams in a K-12 Online Security Competition. In *25th Koli Calling International Conference on Computing Education Research (Koli Calling '25), November 11–16, 2025, Koli, Finland*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3769994.3770022

1 Introduction

In the 2019 iteration of picoCTF, a two-week computer security competition with over 90,000 participants, the top three ranking high school teams presented a set of counter-intuitive properties (especially at the time, before the COVID 19 pandemic): members were all geographically distributed, their collaboration was fully online, and some team members did not even know each other, let alone had collaborated before. Yet, these teams of high school students adopted a set of strategies that allowed them to surpass thousands of other competing teams—many who were co-located—on a series



This work is licensed under a Creative Commons Attribution 4.0 International License.
Koli Calling '25, Koli, Finland
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1599-0/25/11
https://doi.org/10.1145/3769994.3770022

of security engineering tasks. While the research literature on distributed teams may provide hints on the factors that contributed to these successful collaborations [24, 36, 68], little attention has been given to collaboration among *security engineering* teams specifically. In particular, there is scant research on the specific practices of teams that join together online, without much/any prior interaction, to complete short-term security tasks.

Computer security educational competitions, such as picoCTF, translate to real-world security crowd work efforts. In today's era of ever-evolving security threats including generative AI, these community-driven security efforts such as bug bounty programs and penetration testing are critical layers of defense [3]. Companies, through bug bounty programs, invite the public to review their software, and issue payments for reporting bugs. Due to the cost efficiency of such programs [27], the crowd sourced security industry has been dramatically increasing in recent years [3, 15]. However, current crowd-sourced security offerings focus on harnessing work at the individual level (i.e., slicing and distributing work to individuals), which introduces issues of work replication, reduced report quality due to competition, and attrition [95].

Scholars focusing on security crowd work have so far primarily focused on rigorous, quantitative modeling, mainly from an economics perspective [48, 95], to tackle this problem. While economics-inspired approaches are relevant, we argue that the problem is ripe for investigation from a collaborative-work computing education perspective as well, as computer security competitions are designed for security experts and computing students alike. In discussing the future of crowd work, Kittur et al. proposed a framework to scale work to support more complex tasks and increase efficiency, among other gains. However, lacking from these conceptualizations of the future of security crowd work are insights about how security teams operate and what their needs are. Through this study of capture-the-flag (CTF) teams of high school computing students, we explicate several issues relevant to distributed teamwork in computing education, such as strategies developed by students pre-pandemic for effective remote collaboration on computer security tasks, and how technologies impacted students from diverse social identities. In addition, we make connections to current frameworks of understanding security crowd work, arguing that CTF competitions are similarly short and teams often collaborate, form, and self-organize remotely. Additionally, the challenges that teams solve in CTF competitions involve similar skill sets to those needed for security engineering in industry; in fact, bug-bounty programs even run their own CTFs as an educational resource [1, 14].

To explore the dynamics, practices, and needs of high-performing teams of high school students collaborating on computer security engineering tasks, we conducted focus groups with the top five ranking teams from the 2019 picoCTF competition. Our study is exploratory and sought a wide view, from the formation of teams to their motivations, with a focus on the practices that enabled their high performance. Our research questions are as follows:

RQ1: *How, why, and through which assembly mechanisms do high-performing teams form in K-12 computer security competitions?*

RQ2: *What technological needs do these teams have, and how do they address them?*

RQ3: *How do these teams scope and distribute tasks, roles, and responsibilities?*

RQ4: *What motivations and limitations do these teams have?*

RQ5: *What factors and strategies seem to contribute to these teams' performance?*

We report qualitative findings in response to the questions above, and how these high-performing teams of high school computing students organically adopted various strategies which readily fit the more advanced framework of crowd work proposed by Kittur et al., particularly in terms of task decomposition, hierarchical/reputational organization, task assignment, and collaboration.

2 Related Work

2.1 Student Team Formation and Performance

Prior works exploring characteristics of effective teams in cybersecurity competitions is limited, with no work in this area to our knowledge in the K-12 context. At the higher education level, La Fleur et al. conducted a Bayesian analysis of predictors of successful team performance in a collegiate cybersecurity competition. Team organization strategies, such as role specialization, and certain prior experience involving hands-on simulated cybersecurity exercises (as opposed to lecture-based cybersecurity education, for example) were found to predict higher performance [46].

CS education literature focusing on student teamwork has mainly taken place in formal education environments such as upper-level software engineering courses, although CS pedagogies such as the application of POGIL also center around team collaboration and role assignment [45]. Among the challenges reportedly faced by student teams in software engineering courses are poor communication [23, 58, 72, 73], internal conflicts [23, 39, 58], lack of motivation [23, 58] and engagement [39, 72], and difficulty with project management [39, 73]. Recent work involving semi-structured interviews with students who recently participated in a six-week software engineering team project confirmed these existing challenges as well as additional challenges identified from the student perspective, such as mental health challenges [73]. This same study additionally explored characteristics of successful teams, which included maintaining regular communication (with eight out of nine groups studied using Discord [21]), working on tasks together, and investing heavily in task planning [73]. Works investigating the transition of software engineering student teams to fully online work at the start of the COVID-19 pandemic reported that this both exacerbated existing challenges such as team member motivation, as well as creating new ones related to creating effective relationships and the use of non-collaborative software [91].

Studies of team formation strategies and performance in the broader education literature have produced conflicting results regarding instructor-formed versus self-selected student teams [66, 87]. In addition, prior work on self-selecting student teams in higher education outside of CS has shown that self-selection "places students in situations that perpetuate ... barriers to developing diversity skills: stereotyping, prejudice, and discrimination" [13, 64]. That being said, students with smaller existing social networks to draw from were more likely to end up in diverse groups where they could develop critical diversity skills [10, 64]. Within CS education literature, several works comparing self-selected student teams to alternative methods based on performance and student satisfaction

have shown either no significant difference, or that the student-selected teams outperform others [55, 86]. Often, these strategies are combined in formal education environments as all students must be placed in a group (e.g., allowing students to request students they do or do not want to work with, and instructors accommodating these requests to group students with other students who were not requested [73]). Therefore, in addition to focusing on student teams modeling security crowd-work environments rather than software engineering teams (see Section 2.3.3), our context outside of the formal education environment offers the additional novelty of student teams being entirely self-selected, which, to our knowledge, has not been covered in previous work.

2.2 Distributed Teams

In addition to the work cited above on virtual teamwork in an undergraduate software engineering course [91], the challenges of virtual or geographically-distributed teams have been well studied and reviewed in industry, particularly for software engineering teams [5, 29, 71]. Although we are not aware of similar prior research on security teams Abarca et al. conducted a systematic literature review of 2,354 studies on virtual teams between 2015 and 2019 [5], following the review by Gilson et al. which covered work from 2004 to 2014 [29], as well as the review by Powell et al. in 2004 [71]. Below, we describe the main constructs that resulted from these reviews.

Geographic Distribution and Communication: Dispersion reduces the amount of communication within teams, creating difficulty in coordination and awareness, which increases delays [68]. Greater spatial distribution also introduces temporal challenges. For example, less overlap in working hours makes it harder for teams to adjust when unexpected problems arise and makes communication more prone to breakdown [24, 25]. In the case of software engineering teams, Herbsleb and Mockus report that geographical distribution leads to more software failures, but increased awareness, communication, and better work distribution can reduce cross-site delays [35]. Newer studies have also looked at how technologies that facilitate coordination and resource sharing can help distributed teams cope with this dispersion [93].

Trust: Trust, defined as "as an individual's willingness to become vulnerable to the actions of others with the expectations that others will follow through on their commitments, [59, 76]" is an essential component of successful distributed teamwork. In a study of globally distributed development teams, participants described trust as expectations of their colleagues, such as technical competency [7]. Trust is cultivated when teammates fulfill the positive expectations of their team members [70, 77]. Increased trust fosters knowledge sharing and coordination, and leads to greater performance [41, 60, 82].

Types of Tasks and Interdependence: The nature of tasks and their assignment impacts team performance. When members work on tasks with more interdependence, more communication and coordination is needed, particularly if these are unplanned. In software engineering teams, the high amount of interdependence between tasks is among the variables that most influence team performance [35, 47].

Cohesion and Familiarity: Cohesion relates to the sense of unity in a team [5, 18]. Physical distance can make it harder for teammates to get to know (i.e., increase familiarity with) each other [81], but given enough time distributed teams may still become cohesive through online communications [18]. Past studies have found that cohesion leads to better performance [53]. Similarly, knowing how other members work or who knows what ("team awareness") can also increase team performance [22]. As such, prior collaboration, such as "team dating" (e.g., interacting in brief tasks before choosing teams) can help members do better, particularly in ad hoc scenarios [54].

Leadership and Motivations: Team leadership can be an effective tool for increasing motivation and coordination within a team [12]. Past research, however, has found that the lack of face-to-face contact attenuates the positive effects of hierarchical leadership [12, 74]. Some scholars suggest that structural supports (such as fair and reliable reward systems) and shared leadership can complement hierarchical leadership in virtual settings to increase performance [12, 37]. Particularly, the performance gain of sharing leadership across members is based on the premise that it empowers individual members [42]. Hoch and Kozlowski found empirical support for the performance advantages of sharing leadership [38], while Zhu et al. found evidence that leadership behavior by all members in online communities increased members' motivations [96].

2.3 Collaborative Computer Security Work

As previously noted, the related works above mainly focus on teamwork for software engineering, as the prior works on teamwork in CS education and virtual teamwork have been in this context. As the focus of the current study is on students participating in a computer security competition, we now shift to the security work context.

2.3.1 Classifying Computer Security Work. In a 2017 NIST special publication on the creation of a cybersecurity workforce framework, computer security work was divided into seven categories: securely provision (SP), operate and maintain (OM), oversee and govern (OV), protect and defend (PR), analyze (AN), collect and operate (CO), investigate (IN) [65]. Then each of these categories was split into specialty areas [65]. For the purposes of this paper, the specialty areas we consider when discussing security teams in the context of CTFs and bug-bounty programs are the following: test and evaluation (TST, part of SP), vulnerability assessment and management (VAM, part of PR), exploitation analysis (EXP, part of AN), and digital forensics (FOR, part of IN).

2.3.2 Security Teams. Most of the academic work on security teams has focused on computer security incident response teams (CSIRTs). CSIRTs are part of the PR category, and the incident response (CIR) specialty area. Work CSIRTs has focused on: exploring it from an organizational psychology perspective [17], formalizing management capabilities [79], and strategies to develop shared knowledge and increase performance [88]. Beyond CSIRTs, Henshel et al. identify the need for and propose an assessment model for quantifying cyber defense teams' proficiency [34]. Kokulu et al. studied Security Operation Centers (teams of security analysts who monitor, prevent, report, and respond to security attacks) and found that there

are disagreements between managers and analysts which threaten the efficiency and effectiveness of these security teams [44].

2.3.3 Bug Bounties and Crowdsourced Security. Bug bounty programs are programs offered by organizations through which external individuals are given the opportunity to look for security bugs in the organizations' software. Those who find a bug can then submit a report to the organization, and depending on the criticality of the bug, receive commensurate compensation (i.e., a bounty). The core idea is that harnessing the wisdom of the crowd is both economically efficient, and also results in more discovered bugs. Both of these claims have been validated empirically, the former by Finifter et al. and Walshe and Simpson, and the latter by Maillart et al..

Bug bounty programs (BBPs) are arguably the most popular crowdsourced security offering, with over 1400 participating organizations, 450,000 registered "hackers", and more than US$62M paid in bounties in the HackerOne platform alone [32]. However, BBPs suffer from a high number of duplicate submissions (30-40%) due to the competition between researchers. Security researchers may devote weeks to the same security bug only for the first submission to be rewarded [95]. This has caused researchers to quickly flock to new programs, prioritizing higher payouts and easier bugs [56].

2.4 Computer Security Competitions

It can be difficult to engage a diverse population of young students in cybersecurity education [11, 20, 83]. This has led to innovations specifically in this area of CS education such as cyber competitions and gamification [11, 20]. According to a 2023 literature review of cybersecurity educational competitions, capture-the-flag (CTF) competitions are the most common type [11]. While this review details the history of CTFs dating back to a 1996 event hosted by DEFCON, CS education literature on the educational impacts and improvement of such interventions are scarce. In one recent example of such work, Costa et al. found that a CTF competition in Italy with girls age 14-19 increased participants' reported interest in pursuing STEM degrees in higher education. Other prior work from a large CTF in India with participants including both college students and working professionals in CS found gender disparities in CTF competition learning outcomes, differing levels of intrinsic vs. extrinsic motivation, and security subdomain preferences, albeit showing a positive impact for both men and women on motivation and self-efficacy [28].

Capture-the-flag (CTF) competitions are events where teams of computer security experts and/or students compete to showcase their skills across various areas of security, such as memory corruption, web security, and cryptography. In jeopardy-style CTFs, puzzles typically involve finding and exploiting vulnerabilities, reverse engineering binaries, or finding information through digital forensics tools. While originally conceptualized as an entertaining battle of wits between talented hackers, CTFs have become much more. The NSA as well as companies like Google and Meta organize CTF competitions to identify and recruit security talent [2, 9, 26]. In schools and colleges, CTF problems are increasingly used as a way to teach security concepts through hands-on exercises [63], and competitions are organized to foster interest in a field with a shortage of security professionals [40]. Finally, bug bounty platforms have also embraced CTFs to provide education to bug bounty hunters [1, 14]. In short, CTFs have developed into a quintessential part of the education and recruitment of current and future computer security professionals.

3 Study Context

This section describes the picoCTF competition and its participants. We provide a sample challenge from the competition in Figure 1 and describe its solution to provide context on players' skill sets and expertise.

Participants in this study were members of the top five US-based teams of high school and middle school students in the 2019 picoCTF computer security competition, which is organized and run by a research university in the US. The organizers provided the research team with competition data and facilitated the recruitment of members of the top teams. Demographic data collected about participants was limited, especially due to their protected status as minors. However, we do report findings relevant to participants' gender, which was included in data collected from students registering their online accounts to participant in competition (although not required). Of the 25 students included in these top five teams, 21 reported their gender as male, two as female, and two did not complete this field. We do not report gender distributions of individual teams so as not to risk de-anonymizing participants.

While this study and the $14,000 prize pool is restricted to middle/high school students in the U.S., the competition is open and free of cost to participants of all ages from around the world. In 2019, 46,052 registered teams, of which 18,201 had at least one successful submission. The competition was open for two weeks, from September 26th to October 11th, 2019. Teams earned points by completing challenges. In case several teams completed all challenges, their final ranking was determined by the speed at which they solved these challenges. The competition consisted of 124 problems, broken down across 6 categories: general, cryptography, forensics, reverse engineering, binary exploitation, and web exploitation. In the the Global rankings (which includes all registered teams), only 13 teams finished all challenges in the allotted time. While picoCTF provides beginner-friendly introductory challenges, its toughest challenges require deep technical knowledge similar to the knowledge needed to work on real systems. For example, Figure 1 is an example of a technical problem from the competition. To solve this problem, participants have to be comfortable using professional reverse engineering tools, such as Ghidra [1], and be familiar with the GNU C Library (`glibc`) [2], heap management techniques (in this case, `tcache` [51]), and vulnerability exploitation techniques (such as, `tcache` poisoning [85], double free errors [69], or null-byte overflows [4]).

4 Methodology

Our data analysis was based on five focus groups conducted with each of the five top-scoring US-based CTF teams from the picoCTF 2019 competition. We invited all members of these teams (a total of 25 individuals) to participate in the focus groups; 17 members

[1] https://ghidra-sre.org/
[2] https://www.gnu.org/software/libc/

Challenge Name: "zero_to_hero" **Points:** 500 / 34201
Description: "Now you're really cooking. Can you pwn this service?."
Goal: Participants are given access to a binary and two libaries: `libc.so.6` and `ld-2.29.so`. To complete the challenge they must exploit the binary.
Solution: To solve this problem, a participant must bypass the 2.29 `glibc` patch which fixes a double free vulnerability. To achieve this, they need to employ a null byte overflow to change a chunk's size, use this as a double free, and then conduct a `tcache` poisoning attack to overwrite `__free_hook` and succeed in the exploit.

Figure 1: A sample security challenge from the 2019 competition and a summarized description on how to solve it.

participated. We followed a semi-structured interview format, and each focus group was scheduled for 90 minutes. Two researchers were part of each focus group: one researcher asked questions, while the other researcher took notes. Our questions covered eight themes, as described in 4.1. The focus groups were held virtually and the meetings were recorded and then transcribed. The transcriptions were then coded using a grounded theory approach [16, 19], following *open coding* [19] and *axial coding* [19] procedures by three independent coders, as described in 4.2. Not all teams had full representation in the focus groups, which we discuss in Section 6.5.

4.1 Focus Group Script

Our goal was to uncover factors (e.g., practices, technologies, strategies) used by the top-ranking teams to achieve a successful collaboration. Due to the scarce literature, especially in this CTF student team competition context, our interview script was built for breadth. Within these topics, we explored in more depth a particular topic if it seemed to be more salient (e.g., the team recurrently mentioned it) or if the team explicitly emphasized its importance. The list of topics we explored are as follows:

- **Formation**: composition, recruitment, assembly mechanisms (as defined by [33])
- **Technology**: use of technology tools, their purpose, and desired/missing functionality
- **Communication**: media and purpose of members' communications
- **Collaboration**: how, when, and why team members chose to work together
- **Dynamics**: roles, responsibilities, and decision processes (e.g., leadership)
- **Coordination**: task definition, creation, assignment, and review and feedback processes
- **Motivation**: which incentives motivated individuals to join and stay in their teams
- **Limitations**: what limiting factors prevented teams/individuals from succeeding

4.2 Data Analysis

Our analysis was based on five focus groups (one per team) with a total of 17 participants. Three independent coders first listened to all focus group recordings and read each transcript. The second author (who also led the focus groups) then grouped question/answer pairs across the 8 topics used in the interview script, keeping follow-up questions and answers together as appropriate. Transcriptions were then coded following an open coding approach to refine and reorganize our initial themes [16]. We allowed these codes to deviate from the original question categories, creating new categories as appropriate. Our approach follows Strauss' view [19] of grounded analysis where researchers can start from an initial coding frame and continue to refine the code book as they proceed. After a first round of coding, the researchers compared codes and discussed similarities and discrepancies across their codes and interpretations. The codes were then refined based on the discussions. We follow the recommendation by McDonald et al. to not calculate inter-coder reliability because our goal was to identify emergent themes [61]. Following, the coders jointly worked to cluster categories hierarchically (using axial coding), seeking to more narrowly categorize and sub-categorize codes. Through this process, we focused our discussion on action strategies and consequences (e.g., why did a team make a certain decision, and did this decision affect communication or dynamics?). Additionally, we used the subcategories to enable the creation of a comparison table as a mechanism to laterally compare teams' practices and strategies. Lastly, we discussed teams' similarities and differences across each category. We describe our observations in the upcoming results section below.

4.3 Ethics

Our study was reviewed and approved by the Institutional Review Board (IRB) of our institution before any research activities began. As our research involved minors and voice recordings, we contacted the parents/guardians of the participants to obtain their written consent in addition to that of each participant. Prior to beginning the interviews, participants were once again briefed on the study, the data collection and retention policies, and ways to withdraw from the study. Participation was voluntary and no financial incentives were provided for participation.

5 Results

In this section, we present the results of our analysis framed in the context of our research questions. A snapshot of key characteristics of the top five ranking teams in the US is summarized in Table 1.

Placement	Location & Collaboration	Prior Collaboration	Leadership	Task Distribution
1st	Distributed: fully remote	Some	Informal	All specialists with sub-specialists
2nd	Distributed: fully remote	None	Shared	All specialists with sub-specialists
3rd	Distributed: fully remote	None	Shared	All specialists & basic generalists
4th	Mixed: mostly remote	Some	Hierarchical	All specialists and one generalist
5th	Mixed: mostly remote	Some	Informal	Mix of specialists and generalists

Table 1: Performance and collaborative features of the US top five ranking teams

Results are not categorized by research question, but rather by broader themes that arose from our focus groups and often provide insights applicable to several research questions. Each subsection below corresponds to one of these five broader themes, and Table 2 summarizes which research questions are addressed by each theme. With regards to performance (**RQ5**), we observed that many factors seemed to contribute to performance differently as described by the teams. As such, we discuss performance across each subsection, where pertinent. Commonalities across task coordination, dynamics, and collaboration, seem to be among the most influential factors

Theme	RQ1	RQ2	RQ3	RQ4	RQ5
1	X		X		X
2	X		X		X
3		X			X
4	X			X	X
5	X	X		X	X

Table 2: Summary of which broader themes answer each of the five research questions

in performance. Formation, motivation, and limitations were more heterogeneous across teams and individuals, but seemed to have varying impacts on performance across individuals and teams. We report whether a certain factor had positive or negative impact on performance based on participants' explicit descriptions (e.g., "this strategy helped us work faster") and through observation of implicit themes, such as teams describing how they spent too much time looking for information on their chat logs.

5.1 Roles, Task Distribution, and Modularity

Clearly defined roles based on specialties were the cornerstone of each team, confirming prior work on successful CTF team characteristics at the college level [46]. Specialties were divided based on broad categories of problems (e.g., a reverse engineering specialist), which addresses our **RQ3**. Team formation and recruitment centered around these specialties. Because teams adopted a role-based task distribution approach, they recruited members based on specialty to maximize the coverage of tasks; these observations also shed light on our **RQ1**. Each member took ownership of a category of problems based on their specialty. This approach had multiple benefits reported by participants. For example, it increased efficiency by reducing redundant work,and allowed them to tackle more problems simultaneously:

> T1–*"When we were organizing the team each person kind of has a specific category they're going to do. So by keeping the work to each category, we kind of just eliminated the problem of double work."*
>
> T4–*"Yeah, we are we're very good on like, partitioning problems based on each person's expertise."*

Second, each specialist was able to lead their section and determine how and when they needed help from other team members. This is important, participants said, because problems across categories may need different types or levels of involvement from other members that may be hard to define by a non-specialist. For example, a member working on a cryptography problem may benefit most from a teammate finding information on cryptosystems that could relate to the problem at hand, whereas a member looking for a web security vulnerability might benefit more from somebody actively looking for other potential vulnerabilities in the code.

> T3–*"Once we got through the problems that we could do alone, I think we started to work together on some problems...there are challenges where like all of us are looking into it together because we all have foundations and then sometimes it's just that different approaches is what we need."*

Third, it created a sense of accountability within the team, following lessons from education literature on collaborative learning regarding the value of interdependence to foster individual accountability [52]. For example, members described not wanting to fall behind and drag their team down.

Fourth, it allowed teams to be more modular, and thus if a member was unavailable for a competition, a similar specialist could more easily substitute for the missing member.

> T1–*"We have a poll for who wants to participate in the CTF, and then we try to organize teams based on different categories. So each person is better at certain category, and then we make sure there's one of each category on the team so they have everything covered."*

Fifth, it facilitated training of new recruits. A specialist would become the go-to expert for a category, and thus new recruits who are interested in that particular specialty can be trained by the more senior members.

While the specialist model carries several benefits, two scenarios could cause issues: If the competition or problems involve more categories than team members, members may have to take ownership of more than one category, which may become a heavy burden depending on the quantity and difficulty of the problems across the categories they own. Furthermore, if a specialist gets stuck in their category and no one else has any expertise in the area, progress becomes stunted because other members may not be able to offer necessary help. A few teams mitigated these downsides by having members sub-specializing in other categories and/or by each member developing a general understanding of all categories (i.e., basic generalists). That way, to mitigate the first scenario, the team can assign two sub-specialists to a category. When they are done with their main commitments, they can direct their attention to their secondary category.

> T4–*If there's a five member limit, I always want two reversers slash binary exploitators on [the team], and then two web slash forensics people on [the team]. And then the last one is just everything else. Because if you have one person and one of those bigger categories, like web or reversing, it just gets troublesome, and they might get overwhelmed with the amount of stuff they have to do.*

In the second scenario (i.e., a member getting stuck), the specialist would first collaborate with the sub-specialist. If they are unable to solve the problem together, they would then involve the other members. Since the other members are generalists, then they can offer additional insights and also know how to better support (e.g., whether to write code, or what online articles may be related). Thus, while having highly talented individuals in a team is important, having the right team composition and task distribution makes the team significantly more efficient.

> T2–*"We didn't really have anyone that specialized in reversal problems. We all worked together on that. We all bounced ideas around and checked. Because we do have an overlap in skills. That helps a lot, if we're stuck."*
>
> T3–*"We all have some experience with all the topics, like even when someone gets stuck, the rest of us can all help out in some way."*

Observation #1: Teams organize and distribute tasks across roles based on computer security specialties (e.g., web security, cryptography) to reduce redundant work, increase team modularity, distribute accountability, and to allow each member to define how they want their teammates to provide help when needed.

5.2 Past Performance, Trust, and Leadership

Performance in previous competitions was the most important signal that members use to determine the credibility of potential recruits, relevant to our **RQ1**. Within the team, this indicator may also be used implicitly to determine leadership and as a proxy for trust as defined in Section 2 (i.e., "expectation that others will follow through with their commitment"). These results shed more light on our **RQ3**.

> T5– *"The best way to see how someone would do on a CTF is to just look at their performance on previous ones."*

> T1– *"Competing in past CTFs before would be a really good indicator, I think. And of course, specialization in a category that we need more skill in for the team."*

This signal was generally quantified by three factors: the number of problems an individual has solved, the difficulty of the problems solved, and the speed to solve these problems. This information is typically available at the team level, not the individual level when looking to recruit members. Therefore, insights about individual performance were also informally transmitted by word-of-mouth, inferred by an individual's participation in previous CTFs, or their contribution of write-ups (tutorials on how to solve specific problems). This metric, because it is seen as an objective measure of individual abilities, is used when recruiting new members as a predictor of their performance. Additionally, it seems that performance can be used as a mediator for trust between members. That is, because members take ownership of task portions, other members may feel more at ease if they perceive their teammate(s) as more capable, based on their past performance.

We observed that informal leadership across teams seemed to gravitate to members who had higher performance signals. On the other hand, we also observed that leadership was more distributed across teams where all members seemed to have similar performance signals. In fact, finding other people with similar skill levels was ideal in identifying potential teammates.

> T2– *"I guess I would just like play a CTF with you and see like if our, our skills matched and like, if like you could help a lot, I guess."*

Members of T1 mentioned one teammate as being the "unspoken leader." This member had substantial experience participating in other competitions and doing bug bounties. Similarly, T5 also had two members who did not formally designate themselves as leaders, but made a lot of the decision in terms of organization and task distribution. That being said, they reported having rules (even if followed imperfectly) that put mutual respect above optimizing performance.

> T5– *"Yeah we do have like, little rules about like stealing problems, which I have broken a few times, where like, if someone has started a problem you probably shouldn't like, swoop in and steal it and just finish it because you're faster at writing scripts, which is what I did like five times to [teammate], and I feel kind of bad."*

T4 had a similar member as in T1, but in their case they decided to designate this member the team captain. Alternatively, in the case of T2 and T3, all members rated their abilities at the same level. At the same time, both of these teams described fully democratic decision-making processes: "someone has an idea and we all agree or we disagree."

Observation #2: Past CTF performance is used as a strong determinant of future performance and thus, is an important metric used during member recruitment, along with specialty knowledge. An individual's past performance seemingly correlates with their perceived capability. Consequently, members with higher perceived performance or knowledge, may informally raise as leaders unless the perceived performance/abilities is balanced across members.

5.3 Technology Usage and Needs

Apart from traditional voice and text-based communication needs, all teams mentioned the need for a mechanism to keep track of task ownership and progress, which addresses our **RQ2**. All teams used Discord as their primary communication platform. Discord is a communication software which features text and VoIP channels, with rich integration capabilities and fine-grained access control.[3] Apart from text and voice, users can share images, files, and stream their screen with other members of the server. Teams used text channels for asynchronous communications and disjointed conversations on various topics, with the option to directly talk about something using the voice channel.

Task visibility and tracking was a harder problem, and teams experimented with different approaches. One team used Trello, a Kanban-style web-based application where users can make lists, assign roles, and specify deadlines [4]. The team mainly used it to leave notes under each problem, but seemed to not find it very valuable.

> T3– *"[Trello] was okay I think. We didn't use Trello for [a subsequent CTF]. So I don't think it was completely necessary [for the competition]. But it was sort of helpful to keep track of which problems people are working on and also for keeping track of notes so we don't have to keep searching through the Discord history."*

Another team used Google Sheets for the same purposes. The other teams mostly relied on Discord, but often griped about conversations being mixed up, or retrieving information from the chat history. Alternative strategies involved creating a channel for each problem (the team mentioned it did not work well) and creating a channel for each category, which had better results, but was still prone to information retrieval issues. Furthermore, teams also mentioned they lacked a good way of sharing larger files or code snippets.

> T5– *"Okay, so one annoying thing about Discord [for sharing code] is that there's a message limit of 2000*

[3] https://discord.com/
[4] https://trello.com

> *characters... it'll just complain and say you have to upload a file instead. [Another problem is that] I don't think any other platform even has like code highlighting, you can't even put code blocks with highlighting although maybe Slack can do that?"*

Observation #3: Text and voice communication were adequately addressed by existing software. On the other hand, teams struggled with task visibility (i.e., awareness of what each member is working on), tracking (i.e., tracking progress and resources on each task), and resource sharing and retrieval (e.g., code and files). Teams experimented with various software but no tool seemed to suit all their needs.

5.4 Formation, Motivations, and Challenges

Assembly mechanisms, reasons to join together, and familiarity between members were different across all teams: from friends who went to the same school (e.g., T5), to strangers who found each other through CTF-oriented Discord channels (e.g., T3) (**RQ1**). On the other hand, motivations (why members sought to do well) across all teams and members were mostly similar (although ranked differently by each member), which addresses our **RQ4**. Members often cited a combination of incentives: educational or professional development, entertainment, socialization, prizes, and measuring performance. Notably, even in cases where participants reported that they were unsure about pursuing related careers in the future, they expressed enthusiasm about CTFs specifically as a way to learn about computer security topics.

> T5– *"Like, also, I'm similarly probably not going to go into information security, but like, high quality problems are almost a precious resource, especially high quality solvable problems. Frequently in CTFs, I'm forced to learn a new concept, which is one, frankly, cool, and two, like, is usually within my grasp. Like I think [teammate] posted or told me that in 2017, he had to use the Pohlig-Hellman attack. And if you're forced to learn the Pohlig-Hellman attack via a CTF I think like, that's cool. The Pohlig-Hellman attack is cool. So it's stuff like that."*

Aside from personal motivating factors for participating, some interviewees discussed the motivation to join a team as necessary for taking their performance to the next level.

> T1– *"For me, like my biggest hurdle in getting started with really getting into CTF is probably not having a team. So like doing them by myself was all fun and all but I knew I couldn't really get anywhere. Just doing it by myself. So I didn't really consider CTFs too much until I was on [T5]."*

These motivating factors (including assembly) seemed to have little impact on performance and collaboration as expressed by the teams, except for one team (T3). One team mentioned that they formed the team and recruited members with the goal of earning a top score in the competition. When screening recruits, the founding members as well as the prospective recruits explicitly stated that they wanted to "place" in the competition (i.e., enter the top ranking in the score ladder). This common motivation, as described by participants, played a role when preparing and during the competition.

> T3–*I had been growing my skills from last year. And so I thought if I joined a team, I'd have, like they said, a chance of getting up there in the top five. And so when I saw that message reaching out from a strong team, it sounded like a good opportunity.*
>
> T3– *"Yeah, for me, it was mostly the same reasons. I wanted to place and I also saw that they were running [their own CTF]. It showed that they were good. So that also influenced me. I think it has to do with sort of the goal. We were trying to place really well, as a competitive team. "*

Setting the common goal of doing well in the competition seemingly increased their motivation, and potentially their performance as well, as it increased the accountability of each member in achieving that goal.

Observation #4: While motivation across team members was heterogeneous across most teams, it reportedly played a big role in the performance of one team (T3). This suggests that while common motivation might not be a requirement for a successful collaboration, it may enhance team cohesion and/or performance.

5.5 Gender Self-Disclosure and Added Pressures on Women Participants

During the focus groups, the few women who participated voluntarily discussed their social identity self-presentation in the CTF community, which adds another dimension to our **RQ1** and **RQ2** in terms of how anonymity features of online communities shape self-disclosure of participants' identities, especially in the team formation process, as well as **RQ4** as these insights add evidence of continued marginalization of women students in these spaces. CTF competitions, because they are conducted online and communication in larger groups is typically text-based, often allow participants to limit their self-presentation to a mere online handle. Members from two teams mentioned that this anonymity was relieving, since their abilities were not subject to presumptions based on their identity. However, when this anonymity was no longer in place, they felt pressured to compensate based on their identity.

> P0[5]– *"I guess being aware that I'm the only female in a group, might add a bit of pressure I suppose. Like sometimes I think that if I say something strange or if I mess up somewhere, I might be confirming the assumption that females aren't as good at CS, or maybe I might be confirming certain stereotypes about my gender."*
>
> P1– *"I think the online aspect where we're just all like, you know, playing together, nobody knows where you're from definitely helps or at least makes it ([issues of diversity]) less of a problem."*

Observation #5: Women participants reported unique experiences in the competition, such as self-censoring their self-presentation of their identity on online platforms to avoid presumptions of their

[5]We omit mentioning the team of the above two members (P0 and P1) to preserve their privacy.

abilities, and feeling pressured to perform to a higher level to counteract stereotypes based on their identity.

6 Discussion and Future Work

6.1 Students Reported Successful Strategies for Distributed Security Crowd-work

The top-performing student teams in this study developed a variety of strategies to cope with the multiple known challenges related to dispersed collaboration. At the core of these strategies was the usage of roles to determine coordination (**Obs. 1**), communication (**Obs. 1**), recruitment (**Obs. 2**), and technology usage (**Obs. 3**). This role-based approach attenuated many of the known downsides of dispersed teams, such as by reducing interdependencies and the amount of coordination and communication needed. We also found that teams assembled in a variety of ways, through various channels, and for various reasons (**Obs. 4**), but that quantified performance and roles heavily affected the assembly process (**Obs. 2**).

While many of the problems caused by dispersion documented in the context of software engineering might readily explain the issues that security engineering teams may face, it does not seem like the same remedies extend. We find that security teams cope with dispersion challenges differently than their software engineering counterparts. In our study, teams employed a role-based approach based on separate specialties to determine work distribution and collaboration interactions (**Obs. 1**). Such an approach is likely not possible for software engineers without prior planning due to the interdependencies that arise when building software and the lack of clear-cut roles.

Thus, because the nature of the work and its requirements are different, treatments conceptualized in the context of software engineering teams (such as radical collocation [89]) may have a lesser impact on security teams. In fact, two teams in our panel had the possibility to work in the same physical space, yet still opted for asynchronous remote work.

We believe that future team-oriented research for security teams at the industry level is imperative given the continuous growing need for security services, professionals, and teams. While our study takes place in a student competition requiring a breadth computer security knowledge areas, we note that there exist various other types of security teams which remain unexplored, such as red/blue teams and penetration testers, both in industry and in crowd-work contexts, as well as educational settings.

6.2 Unmet Technology Needs Persist

Different practices tend to imply different needs, and consequently call for different supporting tools. Given the lack of academic work on security teams, it is unsurprising that there are no tailored tools for the collaboration needs these teams described. This is evidenced in our study, whereby teams juggled a variety of tools to cope with their task visibility and knowledge sharing needs (**Obs. 3**). In software engineering, real-time shared environments have been a popular approach to improving visibility and mitigating coordination costs [49, 50, 80]. For security teams which mostly focus on reviewing code, existing issue tracking systems (such as GitHub's) might be adequate if the project allows for a priori task segmentation. However, for faster paced environments, such as a newly launched bug-bounty program, non-real-time issue tracking systems may fall short. Because there may not be sufficient information (e.g., which parts of the codebase contain cryptography bugs) nor time for a priori coordination, short-term security teams would likely benefit from real-time systems which provide information on the portions (e.g., code segments) that other analysts might be working on and the types of tasks or bugs they are testing for (e.g., web bugs). Additionally, it is unclear what tools readily and acceptably extend to other tasks beyond code reviews, such as testing remote server deployments. How should tests of applications or configurations be tracked and made visible? Or how should analysts share helpful resources and strategies?

6.3 Student Teams Modeled Future Opportunities in Security Crowd Work.

In the context of security crowd work, and in particularly bug-bounty programs, the current paradigm is focused on harnessing the expertise of various individuals. However, the current system is inefficient, and invites duplicate and invalid reports caused by competition among individuals [95]. Our study cases that short-term security teams are viable and could help mitigate efficiency issues, as evidenced by team strategies to reduce duplicate work. Participants in our study were able to efficiently self-organize relying only on roles and performance (**Obs. 2**), two signals that are already available across bug bounty platforms. Allowing users in bug-bounty programs to self-assemble may allow for a more efficient allocation of users' abilities within a program. Thus, rather than all participants analyzing everything at the same time, crowd workers can spend more dedicated effort in specific sections. Furthermore, the effectiveness of the program is also poised to increase, as crowd workers can now collaborate and share knowledge, potentially increasing the complexity of bugs they are able to catch while also speeding up the process. Finally, we also envision a benefit across crowd-sourced security communities, as team assembly might motivate and allow for more opportunities for professional development and education of newer crowd workers, which have been motivating factors both for our participants (**Obs. 4**), as well as for joining BBPs according to bug-bounty hunters [6]. Naturally, questions of resource pooling or team-level competition may arise in this new paradigm. To further explore this proposition, a field study with bug-bounty hunters in teams should be conducted.

6.4 Gender Dynamics among Youth in Online Computer Security Communities

Women participants in our study saw the anonymity shield granted by online participation as a beneficial feature, echoing findings from recent work including anonymity usage among undergraduate computing students in online course forums [84]. They reported that anonymity allowed them to participate in online CTF communities without being subject to judgments based on stereotypes associated with their identity. These members reported self-censoring their presentation (**Obs. 5**) across these communities or even within *ad-hoc* teams they became part of.

Within teams, women also reported feeling pressured to perform to a higher standard to counteract stereotypes based on their identity(**Obs. 5**)—an experience of women and people from historically

disenfranchised racial groups that is well-documented in social science literature and specifically pertaining to computing disciplines [62, 67, 75, 78, 94]. With regards to members of the CTF community whose identities are over-represented (e.g., white males), prior work gives evidence that individuals choose collaborators who minimize team diversity in education and crowd-source work contexts, although certain interventions have shown improvement [31, 64, 84, 90]. These observations raise important considerations in how to design online communities, particularly in specialized crowd-work settings, to mitigate feelings of exclusion based on social identity. For example, sites like HackerOne or Bugcrowd may adopt evidence-based interventions that have been shown to promote diversity in self-selecting teams (e.g., [90]). Our results also add insight to a previous experiential paper on using CTF problems with high school students, which found a significant difference between male and female students' experiences. The researchers reported that "the male students thought the [CTF] activities were more enjoyable and interesting than female students [40]." Given our results, we believe that this issue may not be due simply to lack of interest, but could have more complicated roots (e.g., related to the pressure to perform at a higher standard to counteract negative stereotypes of women in computing).

6.5 Limitations

Our study is limited to US-based teams, and the experiences of members of these teams who were able to achieve a top-ranking spot. While we were limited in our ability to report demographic data besides gender, which was additionally limited to an optional binary option given to students who registered to participate in the competition and does not encompass all gender identities, we acknowledge that the demographics of this group of students reflects a multitude of structural and societal advantages enabling them to succeed at this level and develop the strategies and technical competencies discussed in this paper (see [57, 75, 94]). This limits the generalizability of our results, and future work is needed to capture practices of a broader sample of computer security students and practitioners, as well as the effectiveness of current strategies for more diverse teams. That being said, we emphasize how the experiences reported by women were unique in this case study of the top performing US-based teams in a large competition, and add to existing documented evidence of how negative gender stereotypes in computing impact youth in online communities surrounding computer security.

Another limitation is that not all team members were able to participate in the focus group sessions. In particular, one team was only represented by one member. We attempted to mitigate this by framing the focus group around collective group practices (i.e., practices that all members took part on and so every member could answer). While we believe this approach allowed us to have a good picture of each team, we note that in some cases having additional responses from each member added nuance to the team's dynamics.

7 Conclusion

We presented a first view on the reported strategies and challenges of top-performing teams of K-12 students participating in a large CTF-style computer security competition, which is designed to educate and recruit the next generation of computer security professionals and model modern security crowd work practices in industry. Major findings include:

- Teams organically developed many organizational strategies to maximize their efficiency and reach a top spot among thousands of competing teams, some of which are not captured in existing literature primarily focusing on software engineering collaboration contexts. For example, all of the teams reported prioritizing member specialization at the core of their decision-making, from formation to task distribution.
- Despite a wealth of literature on the struggles of geographically dispersed teams and particular challenges facing K-12 and undergraduate students in adapting to virtual work, all five teams seemed to be mostly unhindered by the many challenges posed by dispersed collaboration in this competition which took place before the start of the Covid-19 pandemic: The top three teams were geographically dispersed, and the fourth and fifth place teams –despite being in the same geographic area (students at the same school)– opted to conduct most of their work online, collaborating asynchronously. However, teams reported limitations of current technologies to meet their needs for remote collaboration on security tasks.
- Women participants reported poignant examples of members preferring not to disclose their gender lest their competences were called into question—make resoundingly clear the importance of addressing issues of exclusion based on social identity, particularly in collaborative crowd work environments, and explicate certain affordances currently used by individuals to mitigate these effects (e.g., women relying on anonymity features).

As new technologies are shrinking industry needs for software engineering and increasing the need for skills related to computer security [8, 30], it is important that our knowledge in the field of computing education reflect these shifts as well. Thus, this work provides sheds light on several paths forward in the study of team collaboration and development of relevant computer security skills in CS education to reflect the emerging field of crowd-sourced security. We show that this area is ripe with opportunities for team-oriented scholarship and technology development, from the teams' technology usage—adapting and reworking tools lacking necessary features—to better understanding of how our findings carry over to professional bug-hunters, as well as exposing the continued marginalization of young women and other diverse individuals interested in computer security which prevents this field from experiencing the benefits of diverse teams.

Acknowledgments

We would like to thank our collaborators at CMU's CyLab, picoCTF, and the REUSE program. Special thanks to Megan Kearns at picoCTF who helped our team connect with winners of the picoCTF 2019 competition to conduct the focus groups. This work has been supported by NSF award #1852260.

References

[1] 2020. Hacker101 CTF. https://ctf.hacker101.com/.

[2] 2020. NSA Codebreaker Challenge. https://codebreaker.ltsnet.net/resources.

[3] 2024. *8th Annual Hacker-Powered Security Report 2024/2025.* Technical Report. Hacker One. https://www.hackerone.com/resources/reporting/8th-hacker-powered-security-report

[4] 0x00sec The Home of the Hacker. 2017. Null Byte Poisoning - The Magic Byte. https://0x00sec.org/t/null-byte-poisoning-the-magic-byte/3874.

[5] Víctor M Garro Abarca, Pedro R Palos-Sanchez, and Enrique Rus-Arias. 2020. Working in Virtual Teams: A Systematic Literature Review and a Bibliometric Analysis. *IEEE Access* 8 (2020), 168923–168940. doi:10.1109/ACCESS.2020.3023546

[6] Omer Akgul, Taha Eghtesad, Amit Elazari, Omprakash Gnawali, Jens Grossklags, Daniel Votipka, and Aron Laszka. 2020. The Hackers' Viewpoint: Exploring Challenges and Benefits of Bug-Bounty Programs. (2020).

[7] Ban Al-Ani, Matthew J Bietz, Yi Wang, Erik Trainer, Benjamin Koehne, Sabrina Marczak, David Redmiles, and Rafael Prikladnicki. 2013. Globally distributed system developers: their trust expectations and processes. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work* (San Antonio, TX, USA) *(CSCW '13)*. Association for Computing Machinery, 563–574. doi:10.1145/2441776.2441840

[8] Meraj Farheen Ansari, Bibhu Dash, Pawankumar Sharma, and Nikhitha Yathiraju. 2022. The impact and limitations of artificial intelligence in cybersecurity: a literature review. *International Journal of Advanced Research in Computer and Communication Engineering* (2022).

[9] Josh Armour. 2017. Announcing Google Capture the Flag 2017. https://security.googleblog.com/2017/06/announcing-google-capture-flag-2017.html

[10] Hart Research Associates. 2016. Falling Short? College Learning and Career Success. *NACTA Journal* 60, 1a (May 2016), 1–6. https://www.proquest.com/docview/1791668756/abstract/DFC6355E9DBA4F5DPQ/1 Num Pages: 6 Place: Twin Falls, United States Publisher: North American Colleges and Teachers of Agriculture.

[11] Tyler Balon and Ibrahim (Abe) Baggili. 2023. Cybercompetitions: A survey of competitions, tools, and systems to support cybersecurity education. *Education and Information Technologies* 28, 9 (Sept. 2023), 11759–11791. doi:10.1007/s10639-022-11451-4

[12] Bradford S Bell and Steve WJ Kozlowski. 2002. A Typology of Virtual Teams : Implications for Effective Leadership. *Group & Organization Management* 27, 1 (2002), 14–49. doi:10.1177/1059601102027001003

[13] Richard D Bucher. 2014. *Diversity consciousness: Opening our minds to people, cultures, and opportunities* (4 ed.). Pearson.

[14] Bugcrowd. [n. d.]. LevelUp0x07 - Hack Another Day Bugcrowd. {https://levelup.bugcrowd.com/}.

[15] Bugcrowd. 2019. *Priority One: The State of Crowdsourced Security in 2019.* Technical Report. https://www.bugcrowd.com/resources/reports/priority-one-report/.

[16] Kathy Charmaz. 2014. *Constructing grounded theory* (2 ed.). SAGE Publications.

[17] Tiffani R Chen, Daniel B Shore, Stephen J Zaccaro, Reeshad S Dalal, Lois E Tetrick, and Aiva K Gorab. 2014. An organizational psychology perspective to examining computer security incident response teams. *IEEE Security & Privacy* 12, 5 (2014), 61–67. doi:10.1109/MSP.2014.85

[18] Laku Chidambaram. 1996. Relational development in computer-supported groups. *MIS Q.* 20 (1996), 143–165. doi:10.2307/249476

[19] Juliet Corbin and Anselm Strauss. 2007. *Basics of qualitative research: Techniques and procedures for developing grounded theory.* Sage.

[20] Gabriele Costa, Silvia De Francisci, Margherita Renieri, and Serenella Valiani. 2025. Tackling the Gender Gap in Cybersecurity Education. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1.* ACM, Pittsburgh PA USA, 234–240. doi:10.1145/3641554.3701807

[21] Discord Inc. 2020. Discord. https://discord.com

[22] Paul Dourish and Victoria Bellotti. 1992. Awareness and Coordination in Shared Workspaces. In *Proceedings of the 1992 ACM Conference on Computer Supported Cooperative Work* (Toronto, Ontario, Canada) *(CSCW '92)*. Association for Computing Machinery, 107–114. doi:10.1145/143457.143468

[23] Dora Dzvonyar, Lukas Alperowitz, Dominic Henze, and Bernd Bruegge. 2018. Team composition in software engineering project courses. In *Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials.* ACM, Gothenburg Sweden, 16–23. doi:10.1145/3194779.3194782

[24] J Alberto Espinosa and Erran Carmel. 2003. The impact of time separation on coordination in global software teams: a conceptual foundation. *Software Process: Improvement and Practice* 8, 4 (2003), 249–266. doi:10.1002/spip.185

[25] J Alberto Espinosa and Cynthia Pickering. 2006. The Effect of Time Separation on Coordination Processes and Outcomes: A Case Study. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences* (Kauia, HI, USA) *(HICSS'06)*. IEEE. doi:10.1109/HICSS.2006.463

[26] Facebook. 2016. Facebook CTF is Now Open Source! Facebook. https://www.facebook.com/notes/facebook-ctf/facebook-ctf-is-now-open-source/525464774322241/.

[27] Matthew Finifter, Devdatta Akhawe, and David Wagner. 2013. An Empirical Study of Vulnerability Rewards Programs. In *Proceedings of the 22nd USENIX Conference on Security* (Washington, D.C., USA) *(SEC'13)*. USENIX Association, 273–288.

[28] Saneesh P. Francis, Vysakh Kani Kolil, Vipin Pavithran, Indrakshi Ray, and Krishnashree Achuthan. 2024. Exploring gender dynamics in cybersecurity education: A self-determination theory and social cognitive theory perspective. *Computers & Security* 144 (Sept. 2024), 103968. doi:10.1016/j.cose.2024.103968

[29] Lucy L Gilson, M Travis Maynard, Nicole C Jones Young, Matti Vartiainen, and Marko Hakonen. 2015. Virtual Teams Research: 10 Years, 10 Themes, and 10 Opportunities. *Journal of Management* 41, 5 (2015), 1313–1337. doi:10.1177/0149206314559946

[30] Kuldeep Gurjar, Anshika Jangra, Hasnan Baber, Maidul Islam, and Shabnam Abdulkasem Sheikh. 2024. An Analytical Review on the Impact of Artificial Intelligence on the Business Industry: Applications, Trends, and Challenges. *IEEE Engineering Management Review* 52, 2 (April 2024), 84–102. doi:10.1109/EMR.2024.3355973 Conference Name: IEEE Engineering Management Review.

[31] Diego Gómez-Zará, Mengzi Guo, Leslie A. DeChurch, and Noshir Contractor. 2020. The Impact of Displaying Diversity Information on the Formation of Self-assembling Teams. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.* ACM, Honolulu HI USA, 1–15. doi:10.1145/3313831.3376654

[32] HackerOne. 2019. *The Hacker-Powered Security Report 2019.* Technical Report. https://www.hackerone.com/resources/reporting/the-hacker-powered-security-report-2019.

[33] Alexa M Harris, Diego Gómez-Zará, Leslie A DeChurch, and Noshir S Contractor. 2019. Joining Together Online: The Trajectory of CSCW Scholarship on Group Formation. *Proceedings of the ACM on Human-Computer Interaction* 3, Article 148 (2019), 27 pages. doi:10.1145/3359250

[34] Diane S Henshel, Gary M Deckard, Brad Lufkin, Norbou Buchler, Blaine Hoffman, Prashanth Rajivan, and Steve Collman. 2016. Predicting proficiency in cyber defense team exercises. In *IEEE Military Communications Conference* (Baltimore, MD, USA) *((MILCOM '16))*. IEEE, 776–781. doi:10.1109/MILCOM.2016.7795423

[35] James D. Herbsleb and Audris Mockus. 2003. An Empirical Study of Speed and Communication in Globally Distributed Software Development. *IEEE Transactions on Software Engineering* 29, 6 (2003), 481–494. doi:10.1109/TSE.2003.1205177

[36] James D Herbsleb, Audris Mockus, Thomas A Finholt, and Rebecca E Grinter. 2001. An Empirical Study of Global Software Development: Distance and Speed. In *Proceedings of the 23rd International Conference on Software Engineering (ICSE '01)*. IEEE, 81–90. doi:10.1109/ICSE.2001.919083

[37] Pamela J. Hinds and Sara Kiesler. 2002. *Distributed Work.* MIT Press.

[38] Julia E Hoch and Steve WJ Kozlowski. 2014. Leading virtual teams: Hierarchical leadership, structural supports, and shared team leadership. *Journal of Applied Psychology* 99, 3 (2014), 390. doi:10.1037/a0030264

[39] Claudia Iacob and Shamal Faily. 2020. The Impact of Undergraduate Mentorship on Student Satisfaction and Engagement, Teamwork Performance, and Team Dysfunction in a Software Engineering Group Project. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education.* ACM, Portland OR USA, 128–134. doi:10.1145/3328778.3366835

[40] Ge Jin, Manghui Tu, Tae-Hoon Kim, Justin Heffron, and Jonathan White. 2018. Game Based Cybersecurity Training for High School Students. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, MD, USA) *(SIGCSE '18)*. Association for Computing Machinery, 68–73. doi:10.1145/3159450.3159591

[41] Gareth R Jones and Jennifer M George. 1998. The experience and evolution of trust: Implications for cooperation and teamwork. *The Academy of Management Review* 23, 3 (1998), 531–546. doi:10.2307/259293

[42] Bradley L. Kirkman, Benson Rosen, Paul E. Tesluk, and Cristina B. Gibson. 2004. The Impact of Team Empowerment on Virtual Team Performance: The Moderating Role of Face-to-Face Interaction. *Academy of Management Journal* 47, 2 (2004), 175–192. doi:10.5465/20159571

[43] Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The Future of Crowd Work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work* (San Antonio, Texas, USA) *(CSCW '13)*. Association for Computing Machinery, 1301–1318. doi:10.1145/2441776.2441923

[44] Faris Bugra Kokulu, Ananta Soneji, Tiffany Bao, Yan Shoshitaishvili, Ziming Zhao, Adam Doupé, and Gail-Joon Ahn. 2019. Matched and Mismatched SOCs: A Qualitative Study on Security Operations Center Issues. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, 1955–1970. doi:10.1145/3319535.3354239

[45] Clifton Kussmaul. 2012. Process oriented guided inquiry learning (POGIL) for computer science. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education.* ACM, Raleigh North Carolina USA, 373–378. doi:10.1145/2157136.2157246

[46] Claire La Fleur, Blaine Hoffman, C. Benjamin Gibson, and Norbou Buchler. 2021. Team performance in a series of regional and national US cybersecurity defense competitions: Generalizable effects of training and functional role specialization. *Computers & Security* 104 (May 2021), 102229. doi:10.1016/j.cose.2021.102229

[47] Claus W Langfred. 2005. Autonomy and performance in teams: The multilevel moderating effect of task interdependence. *Journal of Management* 31, 4 (2005), 513–529. doi:10.1177/0149206304272190

[48] Aron Laszka, Mingyi Zhao, and Jens Grossklags. 2016. Banishing misaligned incentives for validating reports in bug-bounty platforms. In *European Symposium on Research in Computer Security (ESORICS '16)*. Springer, Cham, 161–178. doi:10.1007/978-3-319-45741-3_9

[49] Thomas D. LaToza, W. Ben Towne, Christian M. Adriano, and André van der Hoek. 2014. Microtask Programming: Building Software with a Crowd. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, HI, USA) *(UIST '14)*. Association for Computing Machinery, 43–54. doi:10.1145/2642918.2647349

[50] Sang Won Lee, Yan Chen, Noah Klugman, Sai R. Gouravajhala, Angela Chen, and Walter S. Lasecki. 2017. Exploring Coordination Models for Ad Hoc Programming Teams. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, CO, USA) *(CHI EA '17)*. Association for Computing Machinery, 2738–2745. doi:10.1145/3027063.3053268

[51] The GNU C Library. [n. d.]. Memory Allocation Tunables (The GNU C Library). https://www.gnu.org/software/libc/manual/html_node/Memory-Allocation-Tunables.html.

[52] Rachel A Lotan. 2003. Group-worthy tasks. *Educational Leadership* 60, 6 (2003), 72–75. Publisher: ASCD ASSOCIATION FOR SUPERVISION AND.

[53] Jeremy S Lurey and Mahesh S Raisinghani. 2001. An Emprical Study of Best Practices in Virtual Teams. *Information & Management* 38, 8 (2001), 523–544. doi:10.1016/S0378-7206(01)00074-X

[54] Ioanna Lykourentzou, Robert E. Kraut, and Steven P. Dow. 2017. Team Dating Leads to Better Online Ad Hoc Collaborations. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work (CSCW '17)*. Association for Computing Machinery, 2330–2343. doi:10.1145/2998181.2998322

[55] Henrik Hillestad Løvold, Yngve Lindsjørn, and Viktoria Stray. 2020. Forming and Assessing Student Teams in Software Engineering Courses. In *Agile Processes in Software Engineering and Extreme Programming – Workshops*, Maria Paasivaara and Philippe Kruchten (Eds.). Springer International Publishing, Cham, 298–306. doi:10.1007/978-3-030-58858-8_31

[56] Thomas Maillart, Mingyi Zhao, Jens Grossklags, and John Chuang. 2017. Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs. *Journal of Cybersecurity* 3, 2 (2017), 81–90. doi:10.1093/cybsec/tyx008

[57] Jane Margolis, Rachel Estrella, Joanna Goode, Jennifer Jellison Holme, and Kim Nao. 2017. *Stuck in the Shallow End: Education, Race, and Computing*. The MIT Press.

[58] Maíra Rejane Marques. 2016. Monitoring: An Intervention to Improve Team Results in Software Engineering Education (Abstract Only). In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. Association for Computing Machinery, New York, NY, USA, 724. doi:10.1145/2839509.2851054

[59] Roger C Mayer, James H Davis, and F David Schoorman. 1995. An Integrative Model of Organizational Trust. *The Academy of Management Review* 20, 3 (1995), 709–734. doi:10.5465/amr.1995.9508080335

[60] Roger C Mayer and Mark B Gavin. 2005. Trust in management and performance: Who minds the shop while the employees watch the boss? *Academy of Management Journal* 48, 5 (2005), 874–888. doi:10.5465/AMJ.2005.18803928

[61] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-Rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. *Proceedings of the ACM on Computer Supported Cooperative Work and Social Computing* 3, CSCW, Article 72, 23 pages. doi:10.1145/3359174

[62] Ebony O. McGee, Derek M. Griffith, and Stacey L. HoustonII. 2019. "I Know I Have to Work Twice as Hard and Hope that Makes Me Good Enough": Exploring the Stress and Strain of Black Doctoral Students in Engineering and Computing. *Teachers College Record* 121, 4 (April 2019), 1–38. doi:10.1177/016146811912100407 Publisher: SAGE Publications.

[63] Jelena Mirkovic and Peter AH Peterson. 2014. Class Capture-the-Flag Exercises. In *USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE '14)*. USENIX Association.

[64] Wayne A. Neu. 2018. Quantitative Evidence of Students' Use of Social Networks and Social Categorization When Self-Selecting Teams. *Journal of Marketing Education* 40, 3 (Dec. 2018), 161–175. doi:10.1177/0273475318757583 Publisher: SAGE Publications Inc.

[65] William Newhouse, Stephanie Keith, Benjamin Scribner, and Greg Witte. 2017. *National initiative for cybersecurity education (NICE) cybersecurity workforce framework*. Technical Report 2017. National Institute of Standards and Technology. 181 pages. doi:10.6028/NIST.SP.800-181

[66] Barbara Oakley, Rebecca Brent, Richard M Felder, and Imad Elhajj. 2004. Turning Student Groups into Effective Teams. *Journal of Student Centered Learning* 2, 1 (2004).

[67] Vidushi Ojha, Leah West, and Colleen M. Lewis. 2024. Computing Self-Efficacy in Undergraduate Students: A Multi-Institutional and Intersectional Analysis. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. ACM, Portland OR USA, 993–999. doi:10.1145/3626252.3630811

[68] Gary M Olson and Judith S Olson. 2000. Distance Matters. *Human-Computer Interaction* 15, 2 (2000), 139–178. doi:10.1207/S15327051HCI1523_4

[69] OWASP. [n. d.]. Doubly freeing memory. https://owasp.org/www-community/vulnerabilities/Doubly_freeing_memory.

[70] Gabriele Piccoli, Anne Powell, and Blake Ives. 2004. Virtual teams: team control structure, work processes, and team effectiveness. *Information Technology & People* 17 (2004). doi:10.1108/09593840410570258

[71] Anne Powell, Gabriele Piccoli, and Blake Ives. 2004. Virtual Teams: A Review of Current Literature and Directions for Future Research. *SIGMIS Database* 35, 1 (2004), 6–36. doi:10.1145/968464.968467

[72] Kai Presler-Marshall, Sarah Heckman, and Kathryn T. Stolee. 2022. Identifying Struggling Teams in Software Engineering Courses Through Weekly Surveys. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*. ACM, Providence RI USA, 126–132. doi:10.1145/3478431.3499367

[73] Kai Presler-Marshall, Sarah Heckman, and Kathryn T. Stolee. 2022. What Makes Team[s] Work? A Study of Team Characteristics in Software Engineering Projects. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*. ACM, Lugano and Virtual Event Switzerland, 177–188. doi:10.1145/3501385.3543980

[74] Radostina K Purvanova and Joyce E Bono. 2009. Transformational leadership in context: Face-to-face and virtual teams. *The Leadership Quarterly* 20, 3 (2009), 343–357. doi:10.1016/j.leaqua.2009.03.004

[75] Yolanda A. Rankin, Jakita O. Thomas, and Sheena Erete. 2021. Black Women Speak: Examining Power, Privilege, and Identity in CS Education. *ACM Transactions on Computing Education* 21, 4 (Dec. 2021), 1–31. doi:10.1145/3451344

[76] Lionel P. Robert. 2016. Monitoring and Trust in Virtual Teams. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work* (San Francisco, CA, USA) *(CSCW '16)*. Association for Computing Machinery, 245–259. doi:10.1145/2818048.2820076

[77] Lionel P Robert, Alan R Denis, and Yu-Ting Caisy Hung. 2009. Individual swift trust and knowledge-based trust in face-to-face and virtual team members. *Journal of Management Information Systems* 26, 2 (2009), 241–279. doi:10.2753/MIS0742-1222260210

[78] William H. Robinson, Ebony O. McGee, Lydia C. Bentley, Stacey L. Houston, and Portia K. Botchway. 2016. Addressing Negative Racial and Gendered Experiences That Discourage Academic Careers in Engineering. *Computing in Science & Engineering* 18, 2 (March 2016), 29–39. doi:10.1109/MCSE.2016.38 Conference Name: Computing in Science & Engineering.

[79] Robin Ruefle, Audrey Dorofee, David Mundie, Allen D Householder, Michael Murray, and Samuel J Perl. 2014. Computer security incident response team development and evolution. *IEEE Security & Privacy* 12, 5 (2014), 16–26. doi:10.1109/MSP.2014.89

[80] Stephan Salinger, Christopher Oezbek, Karl Beecher, and Julia Schenk. 2010. Saros: An Eclipse Plug-in for Distributed Party Programming. In *Proceedings of the ICSE Workshop on Cooperative and Human Aspects of Software Engineering* (Cape Town, South Africa) *(CHASE '10)*. Association for Computing Machinery, 48–55. doi:10.1145/1833310.1833319

[81] Wm David Salisbury, Traci A Carte, and Laku Chidambaram. 2006. Cohesion in virtual teams: validating the perceived cohesion scale in a distributed setting. *SIGMIS Database* 37, 2-3 (2006), 147–155. doi:10.1145/1161345.1161362

[82] John Schaubroeck, Simon SK Lam, and Ann Chunyan Peng. 2011. Cognition-based and affect-based trust as mediators of leader behavior influences on team performance. *Journal of Applied Psychology* 96, 4 (2011), 863. doi:10.1037/a0022625

[83] Ena Selman-Housein and Brian Plancher. 2025. Improving the Representation of Undergraduate Women in Cybersecurity: A Literature Review. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. ACM, Pittsburgh PA USA, 1043–1049. doi:10.1145/3641554.3701811

[84] Mrinal Sharma, Hayden McTavish, Zimo Peng, Anshul Shah, Vardhan Agarwal, Caroline Sih, Emma Hogan, Ismael Villegas Molina, Adalbert Gerald Soosai Raj, and Kristen Vaccaro. 2023. Engagement and Anonymity in Online Computer Science Course Forums. In *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1*. ACM, Chicago IL USA, 48–62. doi:10.1145/3568813.3600121

[85] Cesare Silvio. 2019. Linux Heap TCache Poisoning. https://blog.infosectcbr.com.au/2019/07/linux-heap-tcache-poisoning.html

[86] Yan Hern Ryan Sim, Zhi Zhan Lua, Kahbelan Kalisalvam Kelaver, Jia Qi Chua, Ian Zheng Jiang Lim, Qi Cao, Sye Loong Keoh, and Li Hong Idris Lim. 2023. Experiences and Lessons Learned from Real-World Projects in Software Engineering Subject. In *2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T)*. 173–177. doi:10.1109/CSEET58097.2023.00036 ISSN: 2377-570X.

[87] Leonard Springer, Mary Elizabeth Stanne, and Samuel S. Donovan. 1999. Effects of Small-Group Learning on Undergraduates in Science, Mathematics, Engineering, and Technology: A Meta-Analysis. *Review of Educational Research* 69, 1 (March 1999), 21–51. doi:10.3102/00346543069001021 Publisher: American Educational Research Association.

[88] Julie Steinke, Balca Bolunmez, Laura Fletcher, Vicki Wang, Alan J Tomassetti, Kristin M Repchick, Stephen J Zaccaro, Reeshad S Dalal, and Lois E Tetrick. 2015. Improving cybersecurity incident response team effectiveness using teams-based research. *IEEE Security & Privacy* 13, 4 (2015), 20–29. doi:10.1109/MSP.2015.71

[89] S. D. Teasley, L. A. Covi, M. S. Krishnan, and J. S. Olson. 2002. Rapid software development through team collocation. *IEEE Transactions on Software Engineering* 28, 7 (2002), 671–683.

[90] Federica Vinella, Rosa Mosch, Ioanna Lykourentzou, and Judith Masthoff. 2022. The Impact of Digital Nudging Techniques on the Formation of Self-Assembled Crowd Project Teams. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*. ACM, Barcelona Spain, 265–275. doi:10.1145/3503252.3531298

[91] Alexis Rae Walsh, Sarah E Norris, Nathaniel Blalock, Daniel Patrick Mountain, and Courtney June Faber. 2021. Exploring the team dynamics of undergraduate engineering virtual teams during the rapid transition online Due to COVID-19. In *2021 ASEE Virtual Annual Conference Content Access*.

[92] Thomas Walshe and Andrew Simpson. 2020. An Empirical Study of Bug Bounty Programs. In *IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF '20)*. IEEE, 35–44. doi:10.1109/IBF50092.2020.9034828

[93] Jeffrey Warshaw, Steve Whittaker, Tara Matthews, and Barton A. Smith. 2016. When Distance Doesn't Really Matter: Effects of Geographic Dispersion on Participation in Online Enterprise Communities. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work* (San Francisco, CA, USA) *(CSCW '16)*. Association for Computing Machinery, 335–345. doi:10.1145/2818048.2835237

[94] Alicia Nicki Washington. 2020. When Twice as Good Isn't Enough: The Case for Cultural Competence in Computing. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 213–219. doi:10.1145/3328778.3366792

[95] Mingyi Zhao, Aron Laszka, Thomas Maillart, and Jens Grossklags. 2016. Crowdsourced security vulnerability discovery: Modeling and organizing bug-bounty programs. In *The HCOMP Workshop on Mathematical Foundations of Human Computation* (Austin, TX, USA).

[96] Haiyi Zhu, Robert Kraut, and Aniket Kittur. 2012. Effectiveness of shared leadership in online communities. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (Seattle, WA, USA) *(CSCW '12)*. Association for Computing Machinery, 407–416. doi:10.1145/2145204.2145269