HackEd: A Pedagogical Analysis of Online Vulnerability Discovery Exercises

Daniel Votipka
Tufts University
dvotipka@cs.tufts.edu

Eric Zhang and Michelle L. Mazurek
University of Maryland
ezhang98@umd.edu, mmazurek@umd.edu

***Abstract*—Hacking exercises are a common tool for security education, but there is limited investigation of how they teach security concepts and whether they follow pedagogical best practices. This paper enumerates the pedagogical practices of 31 popular online hacking exercises. Specifically, we derive a set of pedagogical dimensions from the general learning sciences and educational literature, tailored to hacking exercises, and review whether and how each exercise implements each pedagogical dimension. In addition, we interview the organizers of 15 exercises to understand challenges and tradeoffs that may occur when choosing whether and how to implement each dimension.**

We found hacking exercises generally were tailored to students' prior security experience and support learning by limiting extraneous load and establishing helpful online communities. Conversely, few exercises explicitly provide overarching conceptual structure or direct support for metacognition to help students transfer learned knowledge to new contexts. Immediate and tailored feedback and secure development practice were also uncommon. Additionally, we observed a tradeoff between providing realistic challenges and burdening students with extraneous cognitive load, with benefits and drawbacks at any point on this axis. Based on our results, we make suggestions for exercise improvement and future work to support organizers.

I. Introduction

Historically, the security community has used online hacking exercises to provide practical education, exposing participants to a variety of vulnerabilities and security concepts. In these exercises, participants demonstrate their security concept understanding by finding, exploiting, and sometimes fixing vulnerabilities in programs. Exercises offer discrete practice sets that can be undertaken in a modular fashion, similarly to practice problems commonly included at the end of each chapter in mathematics textbooks. In fact, hacking exercises are commonly considered useful educational tools, with security experts often reporting that they rely on them for their education [1], bug bounty platforms directing those interested in security to start with these exercises [2], [3], and a significant amount of recent security-education work focuses on creating new exercises [4]–[10]. Further, prior work has provided some evidence that hacking exercises can provide valuable immediate feedback to learners in academic settings [7], [11]–[14].

However, analysis of hacking exercises as educational tools is limited. First, many studies only consider a *sparse* few exercises [4]–[10], [13], [15], [16], limiting understanding of the broad set of popular exercises. Prior work also focuses on a few specific measures of learning and engagement [5]–[8], [12], [17], making the evidence *narrow*. In particular, learning factors which are difficult to control for and measure are rarely considered. Overall, exercise organizers have limited guidance for building effective exercises, educators do not know which exercises provide the most effective learning, and researchers do not have a broad view of the landscape of current exercises.

As a step toward expanding this analysis, we review online hacking exercises to address two main research questions:

- **RQ1**: Do currently available exercises apply general pedagogical principles suggested by the learning sciences literature? If so, how are these principles implemented?
- **RQ2:** How do exercise organizers consider which principles to implement?

To answer these questions we performed an in-depth qualitative review of 31 popular online hacking exercises (67% of all online exercises we identified). As part of our analysis, we completed a sample of 313 unique challenges from these 31 exercises. We evaluated each exercise against a set of recommended pedagogical principles grounded in learning theory [18], [19]. We base our approach on previous curriculum evaluation efforts [20], tailoring the pedagogical principles we use for applicability to hacking exercises. Further, we interview the organizers of 15 exercises to understand how they consider which principles to implement.

We found that no exercise implemented every pedagogical principle, but most were implemented by at least some exercises, some in unique and creative ways. Notable exceptions include that many exercises do not provide structure to help students organize knowledge, or feedback to guide their progress through learning objectives. Few organizers had considered *metacognition*, i.e., helping students consider what and how much they have learned at a high level. We also found that some pedagogical principles are in tension with each other — such as balancing difficulty with realism — while others are in tension with the competitive origin of many exercises. Finally, we find that community participation brings many benefits, but must be carefully managed to ensure educational structures are maintained. From these results, we distill recommendations for improving exercises and future work to support organizers.

II. Methods

To understand the current landscape of online hacking exercises, we performed a two-phase study: a qualitative review of popular online exercises and interviews with the organizers of these exercises. Here, we discuss how we selected exercises for review, our review process, and our interview protocol.

A. Exercise Selection

There are many kinds of resources available to security students, such as vulnerability write-ups, certifications, academic coursework, and books. To limit our inquiry's scope, we focus

on online educational exercises—commonly recommended by security experts [1]—, which meet the following criteria:

- **Educational** – Because we are evaluating the educational benefit of each exercise, we only include exercises which explicitly state education as a goal. We do not consider competitions, such as the DefCon Qualifiers, whose goal is to identify the "best" hackers.
- **Hands-on** – Exercises must include a hands-on component requiring students to actively practice security concepts. This component could be the central focus of the exercise—as in many CTFs—or auxiliary, e.g., presented after a series of associated lectures.
- **Online and publicly accessible** - We focused on online exercises so we could analyze them by actually participating, rather than making possibly incorrect assumptions based on an offline exercise's description.
- **Popular** – We opted to focus on popular exercises students are most likely to participate in. To estimate a site's popularity, we used its Tranco rank—a secure method for ranking sites based on user visits [21] — . as of October 15th, 2019. Because Tranco only tracks the top one million sites, we used Alexa rankings if no Tranco ranking was available. Each site's rank is given in Table I.[1]

Because we focused on explicitly educational and popular exercises, many of the exercises we reviewed had funding support, either by offering a paid version of the exercise (e.g., HackEDU, HackTheBox, Mr. Code, Vulnhub), receiving funding through a parent company (e.g., Google supports gCTF and the SANS Institute supports GirlsGo CyberStart), or through grant funding (e.g., picoCTF, BIBIFI). As a result, several organizers we interviewed could dedicate time and resources to improving students' educational experience, which is not necessarily common among CTFs run by professionals in their spare time or university student clubs [22].

1) Exercise Identification: To identify exercises meeting our criteria, we first contacted eight security education experts recruited through one author's personal and professional relationships. We asked each to recommend exercises, publicly available lists of exercises, and possible search keywords. Based on their recommendations, we performed Google searches with all possible combinations of "cybersecurity," "computer security," and "security" with "capture the flag," "CTF," and "war games," as well as "hacking exercises." We reviewed the first 10 result pages per query for candidates. We also reviewed curated exercise lists suggested by our experts [23]–[27].

For each exercise and recommendation list identified, we also reviewed the top three similar sites identified by Alexa.com.[2] After this search, the security education experts reviewed our list to identify any missing exercises and add other terms or lists they had previously mentioned. We continued this process until no new exercises were identified, in October 2019.

[1]This ranking indicates a domain's popularity, not the exercise's specific sub-domain, introducing some ambiguity (e.g., gCTF benefits in ranking from its location at withgoogle.com). However, this was not a common problem in our data, so we believe this popularity ranking is a reasonable approximation.

[2]https://www.alexa.com/siteinfo

While almost all the exercises we identified were joinable year-round, many were initially designed as a live, short-term competition. We expected the initial participation context to affect exercise structure, so for comparison purposes, we assigned each exercise to one of two categories:

- **Synchronous** (N=13) – Designed for simultaneous participation over a short time period (i.e.,a few days or weeks). This includes most capture-the-flag (CTF) competitions. Challenges from these exercises are made available after the competition for more students to try at their own pace.
- **Asynchronous** (N=18) – Designed for participation at any time at the student's pace; often referred to as "wargames."

2) Sample Selection: We identified 45 exercises meeting our criteria (18 Synchronous, 27 Asynchronous). To balance completeness with manual effort, we sampled about 66% for in-depth review. To focus on exercises reaching the most participants, we began with the top 30% (by popularity rank) in each group. We then randomly sampled the remaining exercises until we selected about 66% of each group. We include less visited exercises to account for those still growing in popularity. The final list of exercises is given in Table I. Note, some authors are affiliated with BIBIFI, which was randomly selected during this phase. We did not exclude it to ensure representation of attack-defense-style exercises. To expand this category beyond BIBIFI, we purposively added one more exercise (iCTF), and worked with its organizers to enable analysis despite its highly synchronous (not typically joinable at any time) structure, bringing the total set of reviewed exercises to 31.

B. Pedagogical Review (RQ1)

To identify pedagogical principles, we drew on previous efforts to synthesize major theoretical and empirical learning sciences and education research findings into actionable principles [18]. This led us to five core pedagogical principles: *connecting to learners' prior knowledge* [28], [29], *organizing declarative knowledge* [18], *active practice and feedback* [19], [30], *encouraging metacognitive learning* [31], [32], and *establishing a supportive and collaborative learning environment* [19]. These principles are understood to support human learning generally. While there is little evidence specific to security education for these principles [5], [7], [15], they have been found effective in related domains (i.e., various STEM fields including computer science education), so we expect them to apply to security education as well.

To identify actionable dimensions for each principle, we started with 24 dimensions used by Kim and Ko [20] in their similar review of online programming exercises. Two authors then tailored these dimensions through collaborative open coding of five exercises. For example, Kim and Ko considered whether students wrote code during tutorials as a dimension of the *active practice and feedback* principle. We modify this by asking whether students are required to practice exploiting programs and writing secure code. Additionally, Kim and Ko did not consider *establishing a supportive and collaborative learning environment* because online programming tutorials are typically used in isolation. Because we observed that the

communities around exercises were an important factor, we added this principle to our review. This process resulted in 30 total pedagogical dimensions, across the 5 core principles. We discuss each dimension in further detail in Section III.

For each selected exercise, two researchers independently evaluated each exercise qualitatively coding implementation of the pedagogical dimensions. For reporting simplicity, we used three levels: *yes* (●), *no* (○), and *partial* (◐), as shown in Table I. Using an open coding approach, we defined each dimension's levels based on analyzed exercises. That is, the difference between a partial and a complete implementation emerged from our exercise review. In most cases, "yes" indicates frequent dimension implementation across challenges. Conversely, "partial" indicates the dimension was implemented, but only in one or two of challenges, and "no" means the dimension was not implemented at all in the exercise. We used this approach, rather than setting a challenge-percentage threshold, in order to account for variation across dimensions. We ensure consistency using a dual-coder agreement process, described below. We give specific examples of dimension levels in Section III when they differ from this general definition.

For each exercise, we performed a broad review of all website components (e.g., FAQs, initial instructions, challenge categories, forums, additional resource pages, etc.) to understand the information students might view. Next, we completed at least one logical exercise unit (e.g., all challenges in a category or a single specified path through the exercise). If no logical relationship was present, we completed challenges until we reached *saturation*: when we observed no additional pedagogical methods [33, pg. 113-115]. In all cases, we completed at least five challenges.[3] On average, we completed nine challenges per exercise (313 total; median 9.5; maximum 17). Because we focused on information presentation and not difficulty, we followed published challenge walkthroughs when available, allowing us to review complex challenges quickly. Note that we explicitly targeted mostly challenges marked as less difficult by challenge authors, as we expected organizers to provide the most pedagogical support to earlier-stage students; this was confirmed in our organizer interviews. However, we did review multiple challenges rated as more difficult by the organizers in each exercise to ensure a complete view.

Finally, we reviewed our results with the organizers of 14 exercises to determine whether challenges we did not review implemented additional pedagogy (see Section II-C). Only minor updates were made in these reviews.

After establishing our initial codebook, two researchers independently reviewed 20 exercises, comparing results after every five exercises for inter-rater reliability. (In cases where the dimension could be assessed without any judgment decisions, inter-rater reliability was not calculated, as it is unnecessary [34]. For example, when evaluating whether solutions were available, we assigned "yes" if the exercise offered direct links to solutions or we could find them on a google search's first page.) To measure inter-rater reliability, we used Krippendorff's Alpha (α), which accounts for chance agreements [35]. After each round, the researchers resolved coding differences, modified the codebook when necessary, and re-coded previously reviewed exercises. This process was repeated until an α of at least 0.8—the recommended result reliability threshold [35]—was achieved. The remaining exercises were divided evenly between the two researchers. Final α values are given in the first row of Table I.

[3]One exception: We completed the only three free HackEDU challenges.

C. Organizer Interviews (RQ2)

Because we did not review every challenge in each exercise, we offered organizers an opportunity to provide clarifying information. Also, to answer our second research question, we needed additional context from organizers to understand their decision-making process. As such, we reached out to the organizers of all 31 exercises. For BIBIFI, with which two authors are affiliated, we interviewed the exercise's original architect who was not involved with this paper. We gave each organizer a report describing our review and invited them to participate in a 45 minute structured interview or respond to our review via email. Each report gave all pedagogical dimension definitions, our coding for their exercise, and the reasoning behind our decisions. In our report and throughout our interviews, we were careful to emphasize that our goal was to understand their decision-making, not critique it. We made sure to adopt a constructive tone rather than presenting findings in an accusatory manner. We let organizers know we invited and expected disagreements with our evaluation, as there were likely elements or viewpoints we had not considered. Fifteen organizers responded to our report, 13 participated in a video-call interview and 2 answered our questions via email.

In our interviews, we walked organizers through the report and asked whether they agreed with our assessment and if not, why. Based on organizer feedback, we revisited our results, making updates as needed when the organizers pointed us to challenges or other portions of the site we may have missed. Changes were made based on nine of the 15 responses: two changes each for two exercises, and one change each for the other seven. Updates are indicated with a ‡ in Table I.

For dimensions not implemented, we asked organizers if they considered the dimension when building their exercise and if so, why they chose not to implement it. Our interview protocol is given in Appendix A. Because this study component constituted human-subjects research, it was reviewed and approved by our organization's ethics review board. All raw records, including organizers' identifying information, were maintained securely.

To identify themes in organizers' decision-making, we again performed open coding of organizers' reasons not to implement dimensions. Responding organizers are shown in Table I. To establish our codebook (Appendix B), two researchers reviewed three responses together. Then, those researchers independently coded 12 responses, comparing codes after every three until attaining sufficient inter-rater reliability ($\alpha = 0.86$) [35]. The researchers divided the remaining interviews evenly.

D. Limitations

Our study has several limitations inherent to our review and sampling method, and some common to exploratory qualitative research. First, because many pedagogical dimensions have not yet been evaluated specifically in the security education setting, we cannot say which pedagogy are most effective, how they interact, or how effective they are in varying specific contexts (e.g., effect of narrative for simpler vs. harder challenges). Future work is necessary to answer these specific questions, but evidence from other disciplines strongly suggests that implementing each pedagogical dimension is very likely beneficial in general (other things being equal). This paper is instead intended to map the choices made in current exercises, highlighting tradeoffs based on organizers' practical experiences. This can guide future organizers to make intentional choices about pedagogy and future research to consider specific tradeoffs and contexts to evaluate.

Next, it is likely that we did not identify all candidate exercises meeting our stated criteria. Additionally, because we only review a sample of exercises, we may have missed a particularly good implementation of some pedagogical dimension. However, because of our thorough search process and weighting our sample toward more popular exercises, our results are likely representative of most students' experience.

In our pedagogical review, we adopt a conservative approach, checking whether a dimension is implemented, but not whether it is implemented *well*. We did this to broadly evaluate the types of pedagogy considered and establish an initial understanding of the current landscape. However, we cannot make statements about the efficacy of specific approaches. We encourage future work to build on our established roadmap.

Further, there is likely self-selection bias in which organizers agreed to be interviewed. In general, organizers who are more engaged in supporting student learning may be more likely to respond to a request to discuss pedagogy. We also observed anecdotally that organizers who implemented more pedagogical dimensions were more likely to agree to an interview. While this may reflect engagement in pedagogy, it may also indicate that—despite our best attempts to ensure our feedback was positive and constructive—some organizers found our comments or interview request pejorative. In addition, social desirability bias suggests that organizers may (consciously or unconsciously) tailor their responses to appear in the best possible light. To partially mitigate this, we only revised our dimension assessments if organizers identified exercise elements we missed in our initial review, but did not allow organizers to argue for pedagogical dimension redefinition to better suit their exercise. Overall, our findings regarding organizer decision-making should be interpreted within this context, and may reflect a higher-than-average degree of interest in improving student learning. Nonetheless, we believe they provide novel insights into security education and directions for future work.

Finally, in the next section, we give the number of exercises (N) and organizers (O) that demonstrated or expressed, respectively, concepts, to indicate prevalence. If an organizer did not indicate a specific reason for not implementing a pedagogical dimension, this does not necessarily indicate disagreement; instead, they may have simply failed to mention it.

III. Results

Final review results are given in Tables I and II. Each exercise was assessed on all 30 pedagogical dimensions. Exercises are grouped into synchronous and asynchronous, then sorted by popularity. Overall, we found that while some exercises implemented more pedagogical dimensions than others, no exercise implemented all dimensions. Additionally, we observed innovative approaches to education distributed among all exercises. We organize our discussion around the five core principles, considering each evaluated dimension in detail. For brevity, we only discuss the 23 dimensions included in Table I, which exhibited reasonable differentiation between exercises. The remaining 7 dimensions can be found in Appendix C.

A. Connecting to students' prior knowledge

Learning science research shows that people develop new knowledge based on pre-existing knowledge, including facts, perceptions, beliefs, values, and attitudes [28], [29], [66]–[68]. Students interpret new information through their current world view, and they bring a variety of prior experiences into learning.

Students develop understanding through the production of analogies and connections to previously learned concepts—in the same domain or otherwise. The prior knowledge a student brings to a new context can facilitate learning if it is accurate and complete, or hinder learning if not. Therefore, careful consideration of the students' prior knowledge and deep connection to and activation of that knowledge should help students successfully build new knowledge.

Additionally, supporting tailored education positively affects student motivation. If challenges are appropriately tailored to the student, they will be less likely to feel out of their depth, instead growing their confidence in their learning ability [69].

To evaluate how exercises connected to students' prior knowledge, we considered two dimension groups: *personalization* and *utilization*.

1) Personalization: Each student has a unique background, so exercises should adjust challenge presentation and difficulty to account for these differences, or target specific sub-populations [70]. We considered three personalization dimensions we believe (based on prior work in security [1], [71]–[74] and learning science [29], [66]) likely affect learning background: *age*, *educational status*, and *security experience*.

Experience-based personalization was common. Most (N=23) exercises allow some personalization by experience. These exercises used a mix of difficulty indicators, including difficulty labels (e.g., Easy, Medium, Hard) (N=10), the number of other students who have solved the challenge (N=15), and point values (i.e., more points indicate increased difficulty) (N=18). This guides participants to problems appropriate to their experience level, avoiding burnout on problems beyond their reach or boredom with challenges they can easily solve. This student-guided personalization can also give autonomy,

		Personalization (III-A1)			**Utilization** (III-A2)	**Organization** (III-B1)		**Context** (III-B2)					**Actionability** (III-C1)	**Feedback** (III-C2)				**Transfer Learning** (III-D1)		**Support** (III-D2)		**Peer Learning** (III-E1)		**Inclusive Setup** (III-E2)		
	α		1	1	0.8	1		1	1	–	–	–	1	–	1	–	0.9	–	0.8	0.8	–	–	–	0.8		1	1
Exercise	**Rank**[1]	Age	Education	Experience	Subsequent knowledge	Hierarchical	Problem path	Lecture-based	Project-based	Storyline	Real world exercises	Write secure code	Tailored Feedback	Question support	Hints (scaffolding)	Available solutions	When to use	Why to use	Additional materials	Problem-specific materials	Teams	Group discussion	Low extraneous load	Supportive terminology
Synchronous																								
gCTF† [36]	1.4	○	○	◐	◐	◐	◐	○	○	●	○	○	○	●	●	◐	○	○	○	○	○	○	○	●
iCTF† [37]	1.4	●	●	●	○	○	○	○	○	●	◐	●	●	●	◐	○	○	○	○	○	○	◐‡	◐	●
Infosec Institute [38]	14.1	○	○	●	○	○	○	○	○	○	○	○	○	○	●	●	○	○	●	●	○	○	○	○
picoCTF† [39]	149.8	●	●	○	●	●	●	○	○	●	○	○	○	●	●	●	◐	◐	●	●	●	●	●	●
CSAW365 [40]	*1228.1	○	○	●	●	●	○	○	○	○	○	○	○	○	●	◐	○	○	●	○	●	○	○	●
HackEDU [41]	*2014.2	○	○	●	●	●	●	●	○	○	●	◐	○	◐	●	●	●	●	●	●	○	○	●	●
Pwnadventure† [42]	*2364.7	○	○	○	●	◐‡	●	○	○	●	○	○	○	○	○	◐	○	○	○	○	○	○	●	◐
PACTF [43]	*9156.0	●	●	●	●	○	○	○	○	○	○	○	◐	●	●	◐	○	○	○	○	●	●	●	●
Angstrom† [44]	*11708.9	○	●	●	●	●	●‡	○	○	○	○	○	○	●	●	◐	◐	◐	○	○	●	○	●	●
HXP CTF [45]	–	○	○	●	●	○	○	○	○	○	○	○	○	●	◐	◐	○	○	◐	◐	●	●	●	◐
BIBIFI† [46]	–	○	●	○	●	○	○	○	●	○	◐	●	●	●	○	○	●	●	○	○	●	○	○	◐
Pwn College† [47]	–	●	●	●	●	●	●	●	○	○	○	○	◐	◐	◐	●	●	●	●	○	○	○	●	◐
GirlsGo CyberStart† [48]	–	●	●	●	●	○	●	●	○	●	○	○	○	●	●	●	●	●	●	●	○	◐	●‡	●
Asynchronous																								
HackTheBox† [49]	97.3	○	○	●	●‡	○	○	○	○	○	●	○	◐‡	●	◐	◐	○	○	◐	○	●	●	○	◐
HackthisSite [50]	105.1	○	○	●	●	○	●	○	○	●	○	○	◐	●	●	◐	○	○	◐	○	○	●	○	◐
OverTheWire [51]	151.3	○	○	◐	●	○	●	○	○	○	○	○	○	●	●	◐	○	○	◐	◐	○	●	●	●
Root-me.org† [52]	172.7	○	●	●	●	●	◐	○	○	◐	◐	○	◐	●	◐	●	◐	◐	●	●	●	●	○	●
Vulnhub† [53]	175.8	○	○	◐	●	○	◐	○	○	●	◐‡	○	◐	●	◐	●	◐	○	●	○	○	◐	○	●
Hacker101 [2]	330.4	○	○	●	◐	○	○	○	○	○	○	○	○	●	●	◐	○	○	●	○	●	●	○	●
Hellbound Hackers [54]	432.8	○	○	●	●	○	●	○	○	○	●	●	◐	●	●	◐	◐	○	●	◐	○	●	○	◐
Smash the Stack† [55]	966.1	○	○	○	●	●	●	○	○	○	◐‡	○	○	●	●	◐	◐	◐	●	●	○	●	●	○
Microcorruption [56]	*378.8	○	○	○	●	●	●	○	○	●	○	○	○	○	◐	◐	◐	○	○	○	○	○	●	◐
Pwnable [57]	*515.4	○	○	◐	●	○	○	◐	○	○	○	◐	○	●	●	●	◐	◐	○	○	○	●	●	○
Cyber Talents [58]	*528.0	○	○	●	●	○	○	○	○	○	○	○	◐	○	○	◐	○	○	○	○	○	○	○	◐
XSS-Game† [59]	*626.1	○	◐	○	●	●	●	●	○	●	○	○	◐	●	●	◐	◐	◐	○	○	○	◐‡	●	●
Backdoor [60]	*949.1	○	○	●	●	○	○	○	○	○	○	○	○	●	●	◐	◐	◐	●	◐	○	○	○	●
Crackmes.one† [61]	*1011.4	○	○	●	●	○	◐‡	○	○	○	◐‡	○	○	●	○	●	●	●	○	○	○	●	○	●
CTFlearn [62]	*1267.0	○	○	●	●	○	○	○	○	○	○	○	○	●	◐	●	○	○	○	○	○	●	●	●
HackerTest [63]	*1254.5	○	○	○	●	●	●	○	○	●	○	○	○	●	○	◐	◐	◐	○	○	○	○	●	◐
Mr. Code† [64]	*4570.2	●	○	●	●	●	●	●	○	●	○	○	●	○	○	○	○	○	●	○	○	○	●	●
IO Wargame [65]	*7168.8	○	○	○	●	○	●	○	○	○	○	○	◐	●	●	◐	●	◐	○	○	○	●	●	●

[1] Visit rank for the website, in thousands - Alexa if *, otherwise, using Tranco ranking which is less prone to tampering [21].
† An organizer from this exercise was interviewed or responded via email to our review.
‡ Rating was changed based on an interview with the exercise organizer.

TABLE I: Results of our pedagogical review of 31 exercises. Each column indicates whether an exercise implemented the pedagogical dimension fully (●), partially (◐), or not at all (○).

as the student feels more involved in the learning process and likely more motivated to continue participation [69].

Difficulty levels and point assignments are not optimal. These assignments are made based on the best judgment of the organizers, and it can be hard for students to determine what "Easy" or "10pts" means. In our review, we observed multiple cases where more complicated challenges were rated easier or assigned fewer points than less complex challenges in the same exercise (4% of challenges reviewed per exercise, on average). This supports prior findings, which have shown similar difficulty with labeling [16], commonly due to inconsistencies among multiple challenge authors—a common practice in security exercises (N=18). The Vulnhub organizer explained this problem, saying "What you find easy I will find difficult and vice versa… someone new to the industry [might say], 'This is the first time I've seen this, this is really super hard.' Then give it to a seasoned pen tester and he thinks 'I saw that two weeks ago."' This could potentially inhibit students' ability to personalize their learning or their self-confidence. Several exercises appear to understand this problem. Some try to mitigate it by allowing students to rate or comment on exercises (N=5). However, in HackTheBox, challenges can only be rated after solving the challenge, missing feedback from a likely important segment of students, i.e., those stuck on the challenge due to its difficulty. Additionally, two exercises, picoCTF and Root-me.org, only allow students to indicate whether they liked the challenge, which might not correlate with challenge difficulty. Many other organizers provide a dynamic measure of difficulty based on the number of students who have solved the challenge (N=15).

Only two exercises activated prior knowledge. While most exercises implicitly leveraged prior knowledge, only two—Root-me.org and HackthisSite—activated it by drawing students' attention to previously learned concepts they should remember when trying to solve the challenge. Both did this by including a list of related prerequisite knowledge (e.g., HackthisSite listed "some encryption knowledge" as a prerequisite for a Caesar cipher decryption challenge). By pointing to specific prior knowledge, an exercise can help the student select the appropriate knowledge to build on, helping them avoid potential misconceptions [75].

Some exercises required challenges to be solved in increasing complexity order. When exercises did not personalize by experience (N=8), they always had hierarchical problem paths: more complex problems only unlocked after solving lower-level problems. This could get tedious. For example, in picoCTF, more experienced students may be frustrated as they are required to solve several simple problems—designed for new learners—before they can unlock more interesting challenges. When asked the reason for this design, the picoCTF organizers explained it was "just for convenience, since the year-round version is not really any different from the actual competition period." This was a common sentiment among organizers, with all but the XSS-Game organizers stating the lack of experience personalization was intentional to avoid overwhelming new students. XSS-Game's organizers forced students to follow a specific path, only unlocking new challenges when the previous one was solved, to prevent students from jumping in too far and feeling overwhelmed.

Few exercises personalized based on age or education. Nine made explicit mention of the age or education level targeted. The remaining exercises appeared to target university-level students or above. Pwnadventure's organizer explained the exercise was originally designed to be live in conjunction with the finals of a larger university-level CTF—CSAW CTF [76]. While targeting a more educated audience is likely necessary for more complicated concepts, it should be clearly stated—possibly with links to other resources—to help younger, less educated students who might otherwise be deterred from hacking exercises entirely. Pwnadventure's organizer agreed, saying "It wouldn't be a bad idea to give people that context and just say, 'This is how it was designed. So if you find this too hard, that's expected. This was intended for this audience."'

2) Utilization: For *utilization*, we checked if knowledge gained in prior challenges was required to solve later challenges, building on within-exercise prior knowledge.

Exercise designers build clear challenge concept progressions. Almost all exercises (N=29) include some challenges (32% of challenges reviewed on average) whose concepts build on others. As an example, Microcorruption offers a progression across several challenges to teach buffer overflow concepts. One challenge requires the student to disassemble the program and read a hardcoded password string, then the next forces the student to read the assembly code and understand the stack to reconstruct the password. Next, the student must exploit a simple buffer overflow with no mitigations. The progression continues by adding mitigations to complicate exploitation.

In the two cases where subsequent utilization of knowledge was not observed (Infosec Institute, iCTF), all the challenges covered a disparate set of unrelated concepts. This was likely because these exercises had some of the least number of challenges, but chose to cover a breadth of topics. We expect we would have seen subsequent utilization of knowledge if their organizers added additional challenges.

B. Organizing Declarative Knowledge

Another key to effective learning comes in students' ability to transform facts into robust declarative knowledge [18]. To achieve subject mastery, students must go beyond memorizing facts or specific tricks, but also organize the underlying abstract concepts into a structured knowledge base [19], [29], [67], [68]. Prior work comparing experts and novices has found that while experts do tend to know more facts, their biggest improvement comes from the rich structure of their knowledge base [19]. This allows them to improve recall, recognize patterns, make analogies with previously observed scenarios, and identify key differences between contexts, supporting improved knowledge transfer [19], [77]. In an example (drawn from similar challenges across several exercises), after solving a Caesar cipher challenge and then a cryptographic hashing challenge requiring a dictionary attack, the student should identify the common cryptographic weakness of limited key

spaces. They can then apply this abstract concept to solve a future challenge: decrypting text encrypted with RSA where too-small prime numbers were used to generate the key pair.

To support deeper conceptual understanding, exercises can organize challenges according to the concepts taught to make these knowledge structures clear [78], [79]. For this core principle, we considered how the information was organized and the context in which it was presented. We also considered the types of security concepts covered by each exercise, but found little differentiation across exercises. For brevity, we leave reporting on these results to Appendix C.

1) Organization: When asked to organize facts, experts often sort them into hierarchical structures, demonstrating understanding of complex interrelationships [19]. Exercises can use textual and visual cues to help students make these connections. We looked at whether exercises grouped challenges by concept, creating a hierarchy, or highlighted a problem path, showing linear concept progressions through challenges. We considered whether exercises went beyond general categorizations (e.g., crypto, binary exploitation) and presented lower-level structures via explicit cues (e.g., textual, visual, or structural) to help students recognize overarching structure. This differs from *Utilization*, as we consider whether relationships are made explicit, instead of needing to be inferred.

Many exercises lacked explicit structure. Explicit cues, such as challenge names indicating a concept hierarchy or defining a progression through conceptually-related problems, can help students associate individual facts [19], [79]. A majority (N=18) of exercises did not clearly organize problems to group challenges with related concepts together. Similarly, several exercises did not provide a path through more than two to three challenges as an organizing concept guide (N=12).

Interestingly, almost every exercise that relied on crowd-submitted challenges (N=6)—useful for reducing organizer workload while scaling up—did not provide a clear structure. Vulnhub's organizer explained "There is metadata for about a quarter of the challenges on the backend that's saying, this one has file inclusion, this one has an apache vulnerability, whatever. I was going to implement another feature that would take this metadata and help plot it out. Then VMs went up and I have never implemented it…I've just not kept up to date with it because I was going through everything manually and you can't trust each author's opinion." The authors of Root-me.org, by contrast, do present author-provided metadata; their response did not clarify whether this metadata is reviewed (requiring added organizer effort) or not. We note that even in this case, the author-provided categorizations are typically quite broad; fine-grained connections are typically signaled only when a single author develops a set of challenges with incrementing names, showing a progression (45% of challenges reviewed included fine-grained connections).

2) Context: We also reviewed the *context* within which concepts were organized, which can potentially impact information retention and conceptualization [18], [19], [78]. We considered four dimensions. First, whether authoritative content was presented with the challenge (e.g., video or textual lecture). Next, we asked whether the exercise used a goal-driven project approach, which could help with engagement as students see how individual challenges fit within a broader, more realistic context [30]. We also considered whether any overarching story or narrative was provided to connect learning, as people are more likely to remember information when presented in narrative form [80]. Finally, we assessed whether any challenge programs demonstrated realistic complexity. We note that there is significant benefit in simplified challenges, which limit repetitive tasks (e.g., port scanning), focus student attention on specific problems [81], [82], and provide less experienced students an entry point. However, including some realistic challenges could help students see concept relevance, improving intrinsic motivation [69], and could also support knowledge transfer [77] and the development of practical skills [83].

Stories are the only commonly used method. Many exercises included narrative elements (N=12). The GirlsGo CyberStart organizer said they chose to embed each challenge within a narrative to teach: "Why is a hacker or bad guy using this action? As an educator, you're trying not to just state facts and have them absorb them or try a technique and just do it. You want to give context." While narrative was used in fewer than half of exercises, it was by far the most prevalent practice. Few exercises used lectures (N=6), and only one included challenges with sub-tasks that had to be solved to together achieve an overarching goal. Organizers who did not include these contextual elements (O=4) explained they "got in the way of [challenge author] creativity" (Angstrom) (O=2) or do not apply when challenges are"submitted by several different people" (O=2) (Crackmes.one). These organizers agreed that adding context could help students, but would require significant effort and might reduce the number and uniqueness of challenges, with a net-negative effect on learning.

Few exercises included realistic challenges. Few exercises included any challenges representative of real-world-scale programs (N=9). This practice may inhibit learning practical skills for scaling analyses to larger programs. However, many organizers specifically avoid realistic challenges to focus attention on specific concepts, which they considered more important (O=10). Others chose to avoid complexity—and accordingly limit extraneous tasks—because they wanted to make sure their exercise was fun and engaging (O=6). In fact, this is a common educational tradeoff between realistic settings and extraneous load (discussed later in Section III-E2). The gCTF organizers explained "you focus mostly on the problem solving part that gives the players joy…The recon part is required in real world pen testing…, but in some cases, it either will be mostly luck based if you are looking in the right place at the right time, or developing the [necessary] infrastructure will just take most of your weekend." Because of these tradeoffs and the inherent difficulty of building realistic challenges (O=2), it seems likely that realistic challenges should be included purposefully but sparingly.

We note that this was the most commonly updated dimension during interviews (O=3). In all three cases, our result changed from "No" to "Partial," as the exercises included a few realistic

challenges among a large number provided by the community. This may indicate we underestimate the number of exercises providing realistic challenges; however, organizers generally agreed that realistic challenges were rare, meaning the average student would rarely or never encounter them.

While realistic challenges were uncommon, HackEDU used a unique approach to incorporating real-world programs: providing vulnerable programs reproduced from public vulnerability reports on HackerOne, a popular bug bounty platform.

C. Practice and Feedback

Prior work shows students must perform a task to achieve mastery [77], [83], [84]. Through deliberate, active practice, students can translate abstract concepts into practical knowledge. To support this practice, students must also receive tailored feedback to guide learning to specific goals [19]. Without feedback, students may become lost or misunderstand the exercise's learning objective [18], [19], [79]. Therefore, we considered two dimension groups: *actionability* and *feedback*.

1) Actionability: For *Actionability*, we considered the types of tasks exercises ask students to complete. Specifically, whether students had to exploit insecure programs (e.g., perform a buffer overflow or decrypt a weak ciphertext) or write secure programs—from scratch or by patching vulnerable code. For the latter, we did not consider exercises that required students to write programs for exploitation purposes (e.g., to brute-force a cryptographic challenge). We only considered an exercise as meeting this dimension if the code students produced was evaluated for security vulnerabilities. We found that all exercises required students to exploit programs, so we show these results in Appendix C for brevity.

Secure development practice was uncommon in our dataset. Very few exercises (N=5) included challenges asking students to write secure code and two (Hellbound Hackers and Pwnable) only included a few (6% and 12% of reviewed challenges, respectively). Instead, students are left to make the logical jump from identifying and exploiting to preventing a vulnerability without educational support. For example, XSS-Game—explicitly targeted at training developers—has students identify XSS vulnerabilities, but does not include any information regarding the impact of these types of vulnerabilities or how to avoid them. Most organizers agreed that this is because secure development practice is difficult to evaluate (O=8). This included the XSS-Game organizers, who said "the problem is that it's really hard to test that [the vulnerability is] fixed properly... You actually either have to have somebody manually test it, or a really good checker that's checking a ton of edge cases." Other organizers chose not to include secure development challenges because they wanted to limit the scope of their exercise to focus students on exploitation (O=7).

The three exceptions were HackEDU [41], BIBIFI [46], and iCTF [37]. HackEDU used a similar structure to other exercises, asking students to first identify and exploit the vulnerability in sample code. However, students then patch the vulnerable program by following instructions that walked them through how to make the program secure. BIBIFI and iCTF used variations of an attack/defense model. iCTF provided each team with a set of identical vulnerable services running on a shared network. Students were tasked with finding vulnerabilities that they could exploit on other teams' machines (attack) and patch on their own (defense). BIBIFI followed a similar approach, but asked participants to first write their own medium-sized application according to a given specification, considering tradeoffs between security and functionality. Other teams then search for vulnerabilities in the student's code; when found, the original students are asked to patch the identified vulnerabilities.

BIBIFI and iCTF are not unique in their use of the attack/defense model, and we expect we would have seen more examples of secure development practice if more attack/defense exercises were included. Unfortunately, because this model depends on live interaction with other competitors, exercises using this model are typically only available during restricted time periods. We were only able to actively participate in these two, due to direct support of their organizers. However, as both organizers pointed out, the attack/defense model introduces inherent tradeoffs. To facilitate the back-and-forth offensive and defensive actions, motivation throughout the competition is needed. To maintain fair gameplay, this limits possible support, structure, and peer engagement. Also, the live nature of services (i.e., students could patch at any time) introduces logistical hurdles, for example in indicating challenge difficulty. Finally, while attack/defense exercises offer a better option for secure development practice, "coding to the test" is still possible. Other teams have limited time to review and exploit modifications, so students may be incentivized to produce minimal fixes without resolving underlying security problems.

2) Feedback: We considered several potential forms of *Feedback*. The first—expected to be most helpful [85]—was direct, in-context feedback, where guidance is tailored to the student's approach so far, directing them down the "correct path." We also considered whether less tailored feedback was included in the form of static hints or opportunities for students to seek feedback in forums or challenge walkthroughs.

Exercises rarely provided direct feedback throughout. Most exercises only provided direct feedback in the form of a "flag check" (N=19): allowing a student to verify they have identified the correct solution by submitting a random string associated with challenge success. This string matching is likely problematic, as simple typos or copy/pasting issues can lead students to misinterpret rejected submissions as incorrect solutions. This problem is further exacerbated when some exercises (N=4) do not use consistent flag formats, causing students to question whether the string found is actually a flag.

Some challenges provide "correct path" markers. In exercises marked as partial (N=10), some challenges update their output if the student is following the correct path, even if the exploit was not fully successful. For example, in Root-me.org's *Format string bug basic 2* challenge, the program checks for target address modifications. If this address is modified, but not to the correct address, the program outputs "You're on the right track." However, this feedback was sparse within exercises, with only one or two challenges providing it.

Many organizers said providing specifically tailored feedback for each challenge was difficult (O=6). The GirlsGo CyberStart organizer pointed out that for some challenges where exploitation occurs locally, they do not have a way to track student behavior, saying "How would I know what [the student] is staring at?" Instead of providing automated feedback, many organizers opted to provide tailored information in the exercise's forum based on student demand (O=6). Smash the Stack's organizer explained, "They'll either email us and say, 'Hey, we're stuck here' and we'll respond, or they'll join IRC and ask their questions there. Usually, someone, an admin or just other players, will exchange hints." We discuss this forum-based support in more detail in our review of the *Question support* and *Group discussion* dimensions in this section and Section III-E1, respectively. Finally, three organizers said they had not considered providing automated feedback, but agreed that it would be useful to include in future challenges.

BIBIFI and iCTF provide exploit examples. In BIBIFI's break phase and throughout iCTF, students identify and exploit vulnerabilities other teams' services. In BIBIFI, students are notified of successful exploits against their code, demonstrating mistakes they made in development. In iCTF, students monitor network traffic and observe other teams' exploit attempts. Because this likely produces multiple variations on the same exploit, students receive rich feedback on mistakes made during initial development or patching. Unfortunately, in both, there is no feedback more detailed than success or failure. Also, in BIBIFI, because students work on one project, there is no opportunity to practice and receive feedback in varied contexts.

Students can get help when stuck. All but three exercises either allow students to ask the organizers questions or provide static hints to help point students in the right direction (22 do some of both). For example, XSS-Game allows students to reveal hints, which get progressively more informative. In the early challenges in XSS-Game (33% of challenges reviewed), the final hint provides the solution, to help inexperienced students get started. Unfortunately, if hints are not well crafted, they can be misleading and cause the student to consider an incorrect path. For example, in CTFlearn's *Prehashbrown* challenge, the student is given the hint to "login as admin" and is shown a login screen. This might suggest a need to exploit the provided login form. However, the student actually needs to register an account and exploit a SQL injection vulnerability in a search screen provided after login. We did not observe any relation to the admin account when solving this exercise.

In every exercise except BIBIFI, Mr. Code, and iCTF, students can find textual or video instructions for solving some challenges (95% of challenges reviewed on average). In several cases, the exercise provided a few (69% of challenges reviewed on average) directly on their website (N=10). For example, in GirlsGo CyberStart and picoCTF, the organizers provide videos demonstrating how to solve the first few challenges (24% of challenges reviewed on average) to get students started. We also found that walkthroughs were produced organically by participants for most exercises (N=27), even if some organizers already provided some walkthroughs themselves (N=7) or tried to discourage their creation to prevent students from copying solutions (O=3). In BIBIFI and iCTF, students were prohibited from producing write-ups while the competition was live, but encouraged to add them afterwards to provide after-action feedback to other students. Several organizers prided themselves on the support provided by the community and relied on these informal communications to support struggling students (O=5).

D. Encouraging Metacognitive Learning

Metacognitive learning has two main components: students' abilities to predict learning task outcomes and gauge their grasp of concepts [78], [79]. Guiding students to reflect on why their solutions work helps develop deeper conceptual understanding and supports knowledge transfer to new settings [32], [86]–[88]. It also helps students identify knowledge gaps and target further learning [32]. One way exercises could encourage metacognition is to prompt students to verbalize *why* their solution worked, e.g., via a pop-up after submitting a correct solution. This is similar to after-action discussions common in other expert domains, which guide consideration of how lessons learned might apply to other contexts [89].

1) Transfer Learning: To determine whether exercises supported metacognitive learning, we asked whether they taught *how*, *when*, and *why* particular exploits or mitigation techniques should be used. Answering these questions likely helps students apply knowledge learned from the challenge to real-world use.

Few exercises guided transfer beyond the challenge context. While almost all exercises taught *how* to use each concept through hands-on exercises (N=30), very few explicitly explained *when* (N=6) or *why* (N=5) to use a security concept in other settings. In these few cases, the organizers provided authoritative materials (e.g., video lectures or additional reading) around each challenge instructing students on the specific setting details and how approaches should change with new settings. For example, HackEDU and Pwn College provide lecture materials describing progressively more challenging exploit techniques in the face of ever increasing defensive mitigations. While this is a useful tool for learners, it falls short of best practice recommendations for metacognition, which suggest active student engagement [90].

The partial designation was used if it was possible to implicitly determine *when* or *why* a particular concept was needed by comparing similar challenges. However, this is not ideal, as students may need a sufficiently strong *a priori* conceptual understanding to identify the nuanced differences.

Interestingly, this was this was the dimension group organizers most often reported not considering (O=9). As an example, when we explained metacognition to the picoCTF organizer, they said "I don't know if I ever heard of metacognition before… that could really guide us in developing problems that can guide our learners even better."

2) Support: Once students evaluate their grasp on concepts and identify points requiring clarity, they will seek additional information to fill those gaps. Exercises can support students by linking to additional materials beyond the exercise's scope.

Most exercises provided resources for further investigation. A majority of exercises did provide additional resources to some extent (N=17). These materials often took the form of "Useful links" (N=15), sometimes only for a subset of covered concepts (N=4). While these resource lists are useful, students may find it difficult to identify which to follow for a specific question. Some exercises improve on this by providing relevant resource links with each challenge (N=10). For example, HackEDU included links to relevant blog posts in the challenge description. Organizers who did not provide these resources generally believed students were provided enough information to find resources on their own (O=6). The XSS-Game organizers expressed this sentiment, saying "Either from the description of the challenge or the source code, the user should be able to figure out what to learn about."

E. Establishing an Environment Conducive to Learning

Finally, we considered the social environment in which students participate. Social climate (e.g., interactions with other students and educators) has been shown to impact learning generally. A negative environment can hamper student progress, while a positive environment can excite and engage students [19], [91]. By participating in a group setting, students receive mentoring from more senior students, brainstorm possible solutions with peers, and get support and encouragement when stuck [92]. Additionally, the educational environment can have a significant impact on whether students feel "good enough" to participate [91]. If the perceived barrier to entry is high, students may choose not to try. This is especially true for commonly underrepresented populations [93]–[95].

We characterize the environment along two dimension groups: interactions among students (*Peer Learning*) and between the organizers and students (*Inclusive Setup*).

1) Peer Learning: Peer learning intuitively lightens the burden on organizers, as other students can act as a first-line support. More importantly, students can collaboratively develop knowledge (as opposed to being given it by the organizer), producing more robust understanding [92]. Peer learning has also been shown to improve intrinsic motivation, as students who feel that others depend on their participation are more likely to continue in the face of difficulties [19], [69]. To evaluate whether an exercise provided peer-based learning, we considered whether it explicitly encouraged team formation through a provided team-creation feature (i.e., not just allowing team creation out-of-band) and whether there is an online forum created by the organizers for students to discuss challenges.

Exercises help students find community in online forums. Most exercises provided IRC, Slack, or Discord channels or online forums where students could post questions and share their experiences (N=18). For example, picoCTF created a dedicated Q&A forum in Piazza [96] with sections for each problem category for students to post questions as well as view and respond to others' questions. As mentioned in our discussion of *Feedback* and *Question Support*, several exercise organizers said community participation is important for student success (O=8). The HackTheBox organizers explained that they have "a vocal community that everyone chats…in order to help each other to understand challenges and learn." Similarly, the Crackmes.one organizer said "for a newcomer to the platform, if they don't join the Discord, they will not have all the information." When organizers did not provide a discussion forum, students sometimes organized one (these were marked as "Partial" implementations, N=3). Because they are not directly linked by the exercise, in some cases these were only identified through organizer interviews, making our results a lower bound on the number of exercises with a discussion forum. iCTF was marked "Partial" because it did not allow discussion among teams until after the competition to ensure fairness.

Almost all the exercises without a forum were *Synchronous*. Organizers attributed this to their initial competitive design (O=6). For example, the Angstrom organizer explained that during the competition "Everybody's competing against each other." Now that the competition is over, "people are now allowed to collaborate. We should probably add a channel to support that, but we have not." The Mr. Code organizer explained, "People join at different times and learn at different rates," so a forum does not make sense.

Team participation was allowed in most exercises but rarely explicitly supported. A competition setting disincentivizes the close collaboration that may support in-depth learning. Allowing team participation can act as a middle ground. However, few exercises provided support to help students who were not already members of clubs or organizations form teams (N=9). picoCTF provided the clearest example of team support, providing a "team recruitment" channel in their online forum to help students create virtual teams. picoCTF also included a built-in feature for "classrooms," where students could register together in groups with a dedicated scoreboard, as well as resources to help teachers support their classes. Similarly to discussion forums, teams were not supported when students were expected to move at their own pace (O=2) or because organizers wanted to target individual-level competition (O=4). As an example of the latter, the Smash the Stack organizer explained, "We bring people on board to help organize that we see progress through the game rapidly," making individual participation necessary to determine potential new organizers.

2) Inclusive Setup: Finally, we consider each exercise's framing with respect to extraneous load and terminology. Extraneous load is any cognitive challenge required to complete tasks but not directly related to the concepts being taught [82]. Significant extraneous load can cause students to become stuck and quit for reasons unrelated to the challenge's learning goals [69]. Exercise terminology could also affect less experienced students struggling with new concepts. Reassuring terminology may let students know their struggles are expected and that the solution is not beyond them [69]. Conversely, terminology that demeans newcomers may reinforce imposter-like feelings. A "Partial" mark here indicates that we found the terminology used to be neutral: neither supportive nor demeaning.

Extraneous load varied widely. Most exercises introduce some extraneous load, such as determining how to run and reverse engineer a binary compiled for a different OS (e.g.,

installing a virtual machine) (N=11). In another example, Cyber Talents used varying flag formats, making it harder to determine how to correctly submit flags. Many exercises took steps to reduce extraneous load, such as providing browser-based tooling (e.g., wireshark, command line, disassembler) (N=6) or a server with required tools installed (N=5). Perhaps the clearest examples were Microcorruption, which allowed students to perform all required tasks with a browser-based disassembler and debugger, and Pwn College, which included links to binaries pre-loaded in the BinaryNinja cloud service [97].

Extraneous load is not always bad. While reducing extraneous load is helpful for learning—especially for less experienced students—we do not suggest that extraneous tasks be avoided in every case. Typically, these tasks reflect processes students need to understand and perform in a real-world setting, which corresponds to the *real-world* dimension in Section III-B2. In fact, most organizers who did include extraneous load said it was intended to provide a realistic experience (O=6). Perhaps the most extraneous load is introduced by BIBIFI, which requires the development of a medium-sized program with non-security-relevant features. This requires more effort on extraneous tasks, but is intended to be more representative of a real-world programming scenario. The BIBIFI organizer explained "this is just part of the process of building a real system. So that's a tradeoff. We decided to do it because it gives people real experience." Introducing extraneous load should be considered carefully and in the context of a student's learning progression, in conjunction with decisions discussed previously about connecting to prior knowledge and organizing knowledge to provide context (Sections III-A1 and III-B2, respectively).

Most exercises used supportive terminology, but a few marginalized beginners. A majority of exercises included language in their rules or FAQs offering encouragement (N=18). For example, Vulnhub offered several strategies for dealing with "stuck-ness" and Root-me.org suggested a learning path to help new students work up to more complicated problems. This supportive terminology, along with tailoring exercise difficulty to experience (Section III-A1), will likely improve student confidence and engagement [69]. However, some exercises use terminology that marginalizes newer students who might struggle with basic concepts (N=5). This included HackthisSite calling their first challenge the "idiot" challenge and saying "if you can't solve it, don't go crying to anyone because they'll just make fun of you" and Pwn College referring to their easiest challenge level as the "baby" level. The Pwn College organizers explained that "baby" notation is common in the CTF culture, and that using it was intended to give students a point of reference across CTFs. However, they agreed "someone might interpret it negatively, and we will consider this point."

IV. Discussion and Recommendations

Through our online hacking exercise review and interviews with organizers, we found that no exercise implemented every pedagogical principle, but there were many creative approaches taken across the exercise landscape. Overall, our analysis found a few dimensions where exercises showed room for improvement and others where there are clear tradeoffs between principles. We do not expect every organizer to adopt all recommendations or pedagogical methods, but instead hope this paper can serve as a roadmap to help organizers thoughtfully consider their approach, borrow ideas from other exercises, and select the elements that are the right fit for their context.

We identified four key areas for improvement:

- Most organizers did not consider metacognition, and there were few realistic challenges. This may cause difficulties when students try to apply lessons learned to real situations.
- Many exercises lacked clear, explicit structure, which can help students establish a more robust knowledge base.
- Inherent technical challenges led most organizers not to provide secure development practice or tailored feedback.
- Exercises frequently gave students the autonomy to choose a personalized path, but rarely activated prior knowledge to explicitly leverage prior experience for learning.

Interestingly, these findings are mostly unique to security exercises, as compared to Kim and Ko's review of online programming tutorials [20]. Many online coding tutorials provided direct feedback, while this was not common in our exercises. Conversely, online coding tutorials provided little personalization based on student experience, but this was common in security exercises. However, in both settings, support for metacognition was not common. This may suggest metacognition is generally not well known or understood.

We also noticed interesting tradeoffs between principles that should be considered carefully in the design of an exercise.

There is a clear tension between providing realistic challenges and minimizing extraneous load. The more realistic a challenge becomes, the more auxiliary tasks are required. However, this is not a binary decision: There are a range of good options, which should be intentionally and explicitly selected to fit learning objectives and students' current experience level. One potential approach to this tradeoff would be to design exercises so that students move from toy challenges with low extraneous load to more realistic challenges with more extraneous load as they develop expertise, so they leave the exercise prepared to perform similar tasks in the real world. Alternatively, a single "realistic" challenge for each problem type may be sufficient to provide a bridge to real-world scenarios, with all other challenges focused solely on teaching security concepts. Future work is necessary to evaluate potential design choices along this spectrum.

Community participation can have significant benefits, but can be difficult to manage. Many exercises rely on an active community to provide challenges, help assess challenge difficulty, and support student engagement. However, organizers reported that this can create a moderation challenge, as they try to provide overarching structure and context for challenges, make sure new students receive necessary feedback, and ensure a supportive culture in discussions. Therefore, organizers should carefully consider ways to structure community involvement to gain the benefits of broad participation while advancing educational goals. Additionally, from a research perspective,

this dynamic indicates that it is necessary to understand not only how exercises themselves run, but also how the community operates. Future work should investigate the forums and online conversations that have grown around exercises.

Competition can get in the way of education. Competition offers a useful motivational tool; however, it likely also limits avenues of support for less experienced students through collaboration and discussion. We observed that in many cases, organizers leaned toward competition, often simply as an artifact of an initial offering in a synchronous setting. Organizers should be conscious of this dynamic, especially after an exercise is no longer part of a live competition. Prior work has shown competitive environments in STEM education can negatively effect student experience, especially for members of underrepresented populations [69], [98]–[101]. Exercises can focus on providing more support for team-based learning and helping new students join the community.

A. Recommendations

With these findings in mind, we suggest recommendations for exercise organizers and directions for future work.

Support active student engagement in metacognition. Because many organizers did not consider metacognition, a first step would be to apply common techniques from learning sciences to prompt students to consider their learning state. In Section III-D, we provided one example: asking students to describe *why* their solution worked. Another common method asks students to predict an action's result prior to performing the action. For example, students could be prompted—perhaps by updating the target program to provide an initial text prompt—to predict the outcome of an exploitation attempt prior to its execution. This foregrounds students' current target system and exploit mental models and prompts them to compare their prediction to the true outcome. This technique has proven effective in other domains, helping students recognize their own incorrect mental models, allowing them to identify gaps in their understanding and develop deeper conceptual knowledge [102].

Use a graphical syllabus to provide concept structure. A graphical syllabus is a visual representation (e.g., a flow chart) of concepts covered in a course and their relationships [103], [104]. These visualizations have been shown to help students in other domains process and organize information in both traditional and online courses. Exercises could adopt this visual presentation to provide a high-level view of relationships among challenges, as well as to provide a guide through progressions of related challenges. For example, the syllabus could begin as a graph with links among related challenges. As students progress, more information could be revealed showing how the challenges are related (e.g., same vulnerability type). Using a graphical syllabus is also appealing because it fits the gaming motif common in CTFs: it aligns with roadmaps commonly used to demonstrate player progression through game levels.

Incentivize production of educational elements in community submissions. Community-submitted challenges provide a valuable force multiplier. However, because of the number of distinct authors, their organization and the hints and other information they provide can vary widely. This is expected, since adding these additional elements can be tedious relative to the more interesting problem of developing the challenge. This is similar to the well-documented lack of documentation in APIs and open-source software development [105]–[107]. One possible approach is to apply methods from the crowd-documentation literature (popularized by sites like StackOverflow [108]), including curation activities like voting as well as incentives such as reputation scores [109]. Additionally, because there is already a significant amount of community-generated content available in challenge walkthroughs and blog posts, future work could also consider developing tools to support improved knowledge discovery from these sources.

Research is needed to make some dimensions easier to implement. Future research should explore pedagogical dimensions organizers reported as difficult to implement, namely *secure development practice* and *tailored feedback*. The key challenge in secure development practice is evaluating patched codes' security. Manual analysis is time consuming and does not scale [110]. However, static testing may enable "coding to the test" without fixing underlying issues, learning the wrong lessons. Future work could measure static testings' impact on actual learning, as well as proposing and testing mechanisms to elicit security review from other students. This could allow manual evaluation and scale with student population size, while providing additional exploitation practice.

Current approaches to tailored feedback rely on developers writing challenge code to include feedback at key points. This one-off approach is difficult to execute and does not scale well.Instead, future work should investigate generalizable methods to help students understand the target program's execution under an exploit attempt, possibly by developing new visualizations or using machine learning techniques to identify patterns in successful versus unsuccessful exploits [111]–[113].

V. Related Work

Our pedagogical review gives the first comprehensive view of the online hacking exercise landscape; however, there has been significant prior work considering security education. In this section, we review and compare our work to prior research.

Guidelines for building hacking exercises. Educators and practitioners have long recognized the benefit of hands-on practice for computer security education, suggesting the inclusion of hacking competitions into the academic pipeline [114]–[116]. This has led several researchers to propose exercise development guidelines to teach educators how to build these types of exercises and improve educational outcomes [70], [117]–[119]. While many of these guidelines provide limited recommendations for specific pedagogy, Pusey et al. provide suggestions for tailoring challenges to student prior experience [70] and establish a supportive environment for underrepresented populations [119]. In our work, we not only consider a broader range of pedagogical dimensions, but also evaluate whether—and why not—these are applied in practice.

New exercises to address specific pedagogy. Several researchers have proposed novel educational interventions

to implement and evaluate particular pedagogical principles. Several researchers have incorporated elements of peer-based instruction, having students participate in teams of varying experience levels and encouraging collaboration and discussion among and between teams [4], [5], [7], [15]. Other exercises go beyond the traditional program exploitation challenges. This includes challenging students to design and implement secure systems [9], [14], [120] and perform penetration testing, navigating through sophisticated network topologies and pivoting from machine-to-machine [10]. Reed et al. evaluate the value of presenting challenges in the context of an overarching narrative, finding students were more likely to complete challenges associated with a storyline [121]. Chung and Cohen outline challenges related to extraneous load (e.g., difficulty getting technology set up to be able to participate) and propose technical solutions to limit initial student burden [16].

Significant effort in this space has focused on evaluating and improving challenge difficulty, to provide feedback and ensure challenges are appropriate for learner experience. Chung and Cohen reflect on years of experiences running the CSAW CTF [76], highlighting the importance of quality assurance in challenge development to ensure appropriate difficulty and feedback within challenges [16]. Owens et al. introduced more easy and medium difficulty challenges into picoCTF [122], to provide a more gradual difficulty slope for beginning students [7]. They found this increased student engagement and reduced participant dropout over previous years of the exercise. Several researchers have added time-on-task tracking for each challenge to compare student behavior (time spent working on a challenge), student-reported feedback, and original assigned challenge difficulty [6], [121], [123], using this data to tailor future exercise progressions and challenge difficulty ratings. Maennel et al. further suggested this information could be used by organizers in real-time to identify unintentionally difficult challenges and support struggling students [123].

Each of these studies have shown the benefit of particular pedagogical principles in a given context. Our work takes these principles and asks whether they are applied broadly across the exercise ecosystem and therefore impacting student education.

Broad exercise educational reviews. While much of the literature in security education has focused on individual exercises, there has been some research focused on the ecosystem more broadly. Tobey et al. studied engagement among beginning students in three exercises in the National Cyber League, finding that experienced students are more likely to be engaged and continue participation [17]. However, the authors do not indicate reasons for this lack of engagement. By reviewing the way exercises are organized, we hope to provide some indication insights into how the exercises themselves might impact student participation trends.

Karagiannis et al. reviewed 4 open source exercise deployment platforms to evaluate their usability with respect to setup and administration [124]. We ask an orthogonal question, focusing on students' experiences with specific exercises, not educators' experiences setting up exercises generally.

Burns et al. provide an extensive review of 3600 challenges, outlining the concepts covered and developing a framework to assess the difficulty of each [11]. This work offers a useful complement to our own, as it investigates *what content* exercises teach and we evaluate *how* they teach.

Finally, Taylor et al. reviewed the organization and structure (e.g., whether content is dynamic or static, whether the exercise is open source) of 36 CTFs. Our work provides additional depth to this survey as we consider how specific implementation details impact exercises' educational characteristics.

Other studies of security education. In addition to these hands-on hacking exercises, other work has proposed a variety of security-related educational interventions. These interventions employ some pedagogical principles we discuss. Multiple researchers have proposed and evaluated adding secure development education into the developer's daily workflow [125]–[127]. Whitney et al. incorporate security nudges into the IDE, providing security context as developers write code [125]. Weir et al. take a Participatory Action Research approach, embedding a security researcher in the development team to support security decision-making and evaluate this approach's effect over time [126]. Similarly, Poller et al. evaluate the impact of third-party security reviews on security behaviors over time [127]. Other researchers have suggested narrative-based education for computer security. Sherman et al. and Rowe et al. present case studies around exploited systems and have students discuss the cause and potential mitigations [13], [128]. Blasco and Quaglia had students discuss attacks and defenses portrayed in fictional scenarios from popular culture (e.g., *Star Wars: Rogue One*) [129]. Other researchers have had students share stories of relevant experiences [130], [131]. Denning et al. developed a tabletop card game designed to expose participants to general security problems and adversarial thinking through its overarching storyline [132]. Relatedly, Frey et al. developed a tabletop game in which players defend cyber-physical infrastructure, to help players reflect on security-relevant decision processes and strategies [133].

VI. Conclusion

We described a qualitative review of 31 online hacking exercises, combined with interviews with 15 organizers of those exercises, to evaluate the ecosystem as an educational tool. We found that many pedagogical principles were commonly instantiated across the ecosystem, often in thoughtful and creative ways. No exercise, however, embodied all dimensions examined. We identified several situations where organizers must typically make tradeoffs among principles, as well as ways exercise origins in competition can be detrimental for education. Building on these results, we suggested recommendations including adding support for metacognition, adopting graphical syallabi, and incentivizing community members to contribute to educational aspects of the challenges they develop.

Acknowledgments

We thank Rock Stevens, Jordan Wiens, Dave Levin and the anonymous reviewers for their helpful feedback. This research was supported by NSF grant CNS-1801545.

References

[1] D. Votipka, R. Stevens, E. M. Redmiles, J. Hu, and M. L. Mazurek, "Hackers vs. testers: A comparison of software vulnerability discovery processes," *Proc. of the IEEE*, 2018.

[2] HackerOne, "Home | hacker 101," HackerOne, (Accessed 05-21-2020). [Online]. Available: https://www.hacker101.com/

[3] S. Houston, "Researcher resources - how to become a bug bounty hunter," Bugcrowd, 2016, (Accessed 05-21-2020). [Online]. Available: https://forum.bugcrowd.com/t/researcher-resources-how-to-become-a-bug-bounty-hunter/1102

[4] N. Backman, "Facilitating a battle between hackers: Computer security outside of the classroom," in *In Proc. of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE ?16. New York, NY, USA: ACM, 2016, p. 603?608. [Online]. Available: https://doi.org/10.1145/2839509.2844648

[5] J. Mirkovic and P. A. H. Peterson, "Class capture-the-flag exercises," in *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*. San Diego, CA: USENIX Association, Aug. 2014. [Online]. Available: https://www.usenix.org/conference/3gse14/summit-program/presentation/mirkovic

[6] J. Vykopal and M. Barták, "On the design of security games: From frustrating to engaging learning," in *2016 USENIX Workshop on Advances in Security Education (ASE 16)*. Austin, TX: USENIX Association, Aug. 2016. [Online]. Available: https://www.usenix.org/conference/ase16/workshop-program/presentation/vykopal

[7] K. Owens, A. Fulton, L. Jones, and M. Carlisle, "pico-boo!: How to avoid scaring students away in a ctf competition," 2019. [Online]. Available: https://cisse.info/pdf/download.php?file=CISSE_v07_i01_p21_pre.pdf

[8] A. Doupé, M. Egele, B. Caillat, G. Stringhini, G. Yakin, A. Zand, L. Cavedon, and G. Vigna, "Hit 'em where it hurts: A live security exercise on cyber situational awareness," in *Proceedings of the 27th Annual Computer Security Applications Conference*, ser. ACSAC ?11. New York, NY, USA: Association for Computing Machinery, 2011, p. 51?61. [Online]. Available: https://doi.org/10.1145/2076732.2076740

[9] W. Du, "Seed: Hands-on lab exercises for computer security education," *IEEE Security Privacy*, vol. 9, no. 5, pp. 70–73, 2011.

[10] K. Bock, G. Hughey, and D. Levin, "King of the hill: A novel cybersecurity competition for teaching penetration testing," in *2018 USENIX Workshop on Advances in Security Education (ASE 18)*. Baltimore, MD: USENIX Association, Aug. 2018. [Online]. Available: https://www.usenix.org/conference/ase18/presentation/bock

[11] T. J. Burns, S. C. Rios, T. K. Jordan, Q. Gu, and T. Underwood, "Analysis and exercises for engaging beginners in online CTF competitions for security education," in *2017 USENIX Workshop on Advances in Security Education (ASE 17)*. Vancouver, BC: USENIX Association, Aug. 2017. [Online]. Available: https://www.usenix.org/conference/ase17/workshop-program/presentation/burns

[12] K. Qian, D. Lo, H. Shahriar, L. Li, F. Wu, and P. Bhattacharya, "Learning database security with hands-on mobile labs," in *2017 IEEE Frontiers in Education Conference (FIE)*, 2017, pp. 1–6.

[13] D. C. Rowe, B. M. Lunt, and J. J. Ekstrom, "The role of cyber-security in information technology education," in *Proceedings of the 2011 Conference on Information Technology Education*, ser. SIGITE ?11. New York, NY, USA: Association for Computing Machinery, 2011, p. 113?122. [Online]. Available: https://doi.org/10.1145/2047594.2047628

[14] G. Fraser, A. Gambi, M. Kreis, and J. M. Rojas, "Gamifying a software testing course with code defenders," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE ?19. New York, NY, USA: Association for Computing Machinery, 2019, p. 571?577. [Online]. Available: https://doi.org/10.1145/3287324.3287471

[15] J. Mirkovic, A. Tabor, S. Woo, and P. Pusey, "Engaging novices in cybersecurity competitions: A vision and lessons learned at ACM tapia 2015," in *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*. Washington, D.C.: USENIX Association, Aug. 2015. [Online]. Available: https://www.usenix.org/conference/3gse15/summit-program/presentation/mirkovic

[16] K. Chung and J. Cohen, "Learning obstacles in the capture the flag model," in *Proceedings of the 1st USENIX Summit on Gaming, Games, and Gamification in Security Education*, ser. 3GSE '14. San Diego, CA: USENIX Association, 2014. [Online]. Available: https://www.usenix.org/conference/3gse14/summit-program/presentation/chung

[17] D. H. Tobey, P. Pusey, and D. L. Burley, "Engaging learners in cybersecurity careers: Lessons from the launch of the national cyber league," *ACM Inroads*, vol. 5, no. 1, p. 53?56, Mar. 2014. [Online]. Available: https://doi.org/10.1145/2568195.2568213

[18] J. D. Bransford, A. L. Brown, and R. R. Cocking, *How people learn: Brain, mind, experience, and school: Expanded edition*. National Academies Press, 2000.

[19] S. A. Ambrose, M. W. Bridges, M. DiPietro, M. C. Lovett, and M. K. Norman, *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons, 2010.

[20] A. S. Kim and A. J. Ko, "A pedagogical analysis of online coding tutorials," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE ?17. New York, NY, USA: Association for Computing Machinery, 2017, p. 321?326. [Online]. Available: https://doi.org/10.1145/3017680.3017728

[21] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, ser. NDSS 2019, Feb. 2019.

[22] B. Carlisle, M. Reininger, D. Fox, D. Votipka, and M. L. Mazurek, "On the other side of the table: Hosting capture the flag (ctf) competitions," in *Proceedings of the 6th Workshop on Security Information Workers*, ser. WSIW '20. Virtual: USENIX Association, 2020. [Online]. Available: https://wsiw2020.sec.uni-hannover.de/downloads/WSIW2020-On%20the%20Other%20Side%20of%20the%20Table.pdf

[23] J. Wiens, "Practice ctf list," May 2019, (Accessed 05-21-2020). [Online]. Available: https://captf.com/practice-ctf/

[24] S. Langkemper, "Practice your hacking skills with these ctfs," December 2018, (Accessed 05-22-2020). [Online]. Available: https://www.sjoerdlangkemper.nl/2018/12/19/practice-hacking-with-vulnerable-systems/

[25] Y. Shoshitaishvili, "zardus/wargame-nexus: A sorted and updated list of security wargame sites," April 2020, (Accessed 05-22-2020). [Online]. Available: https://github.com/zardus/wargame-nexus

[26] CTFTime, "Ctftime.org / all about ctf (capture-the-flag)," CTFTime, 2017, (Accessed 06-08-2017). [Online]. Available: https://ctftime.org

[27] A. Ruef, E. Jensen, N. Anderson, A. Sotirov, J. Little, B. Edwards, M. W, D. D. Zovi, and M. Myers, "Ctf field guide," Trail of Bits, (Accessed 05-21-2020). [Online]. Available: https://trailofbits.github.io/ctf/

[28] J. Piaget, *Success and understanding*. Routledge, 1978.

[29] J. Piaget and M. Cook, *The origins of intelligence in children*. International Universities Press New York, 1952, vol. 8, no. 5.

[30] K. A. Ericsson, R. T. Krampe, and C. Tesch-Römer, "The role of deliberate practice in the acquisition of expert performance." *Psychological review*, vol. 100, no. 3, p. 363, 1993.

[31] H. J. Hartman, "Metacognition in teaching and learning: An introduction," *Instructional Science*, vol. 26, no. 1/2, pp. 1–3, 1998. [Online]. Available: http://www.jstor.org/stable/23371261

[32] J. H. Flavell, "Metacognitive aspects of problem solving," in *The nature of intelligence*, L. B. Resnick, Ed. Lawrence Erlbaum Associates, 1976.

[33] K. Charmaz, *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*. SagePublication Ltd, London, 2006. [Online]. Available: http://www.amazon.com/Constructing-Grounded-Theory-Qualitative-Introducing/dp/0761973532

[34] N. McDonald, S. Schoenebeck, and A. Forte, "Reliability and inter-rater reliability in qualitative research: Norms and guidelines for cscw and hci practice," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, Nov. 2019. [Online]. Available: https://doi.org/10.1145/3359174

[35] A. F. Hayes and K. Krippendorff, "Answering the call for a standard reliability measure for coding data," *Communication methods and measures*, vol. 1, no. 1, pp. 77–89, 2007. [Online]. Available: http://dx.doi.org/10.1080/19312450709336664

[36] G. Vigna and A. Doupé, "ictf," UC Santa Barbara, 2020, (Accessed 05-27-2020). [Online]. Available: https://g.co/ctf

[37] Google, "Google ctf 2019," Google, 2019, (Accessed 11-30-2020). [Online]. Available: https://ictf.cs.ucsb.edu/

[38] I. Institute, "n00bs ctf labs," InfoSec Institute, (Accessed 05-27-2020). [Online]. Available: http://ctf.infosecinstitute.com/index.php

[39] P. P. of Pwning, "picoctf," Carnegie Mellon University, (Accessed 05-27-2020). [Online]. Available: https://picoctf.com/

[40] N. O. Lab, "Csaw365," CSAW, (Accessed 05-27-2020). [Online]. Available: https://365.csaw.io/

[41] HackEDU, "Hackedu," HackEDU, (Accessed 05-27-2020). [Online]. Available: https://www.hackedu.com/

[42] V. 35, "Pwnadventure sourcery," Vector 35, (Accessed 05-27-2020). [Online]. Available: https://sourcery.pwnadventure.com/

[43] PACTF, "Pactf," PACTF, (Accessed 05-27-2020). [Online]. Available: https://2019.pactf.com/

[44] angstromCTF, "angstromctf," angstromCTF, (Accessed 05-27-2020). [Online]. Available: https://angstromctf.com/

[45] HXP, "Hxp ctf," hxp, (Accessed 05-27-2020). [Online]. Available: https://2018.ctf.link/

[46] M. C. Center, "Build it break it fix it," University of Maryland, (Accessed 05-27-2020). [Online]. Available: https://builditbreakit.org/

[47] Y. Shoshitaishvili and C. Nelson, "pwn.college," Arizona State University, (Accessed 05-27-2020). [Online]. Available: https://pwn.college/

[48] S. C. Institute, "Girls go cyberstart," SANS Cybersecurity Institute, (Accessed 05-27-2020). [Online]. Available: https://girlsgocyberstart.org/

[49] H. T. Box, "Hack the box," Hack The Box, (Accessed 05-27-2020). [Online]. Available: https://www.hackthebox.eu

[50] HackThisSite, "Hackthissite," HackThisSite, (Accessed 05-27-2020). [Online]. Available: https://www.hackthissite.org/

[51] OverTheWire, "Overthewire," OverTheWire, (Accessed 05-27-2020). [Online]. Available: https://overthewire.org/wargames/

[52] R. Me, "Root me," Root Me, (Accessed 05-27-2020). [Online]. Available: https://www.root-me.org/

[53] g0tmi1k, "Vulnhub," Vulnhub, (Accessed 05-27-2020). [Online]. Available: https://www.vulnhub.com/

[54] H. Hackers, "Hellbound hackers," HellBound Hackers, (Accessed 05-27-2020). [Online]. Available: https://www.hellboundhackers.org/

[55] S. the Stack Wargaming Network, "Smash the stack," Smash the Stack Wargaming Network, (Accessed 05-27-2020). [Online]. Available: http://smashthestack.org/

[56] N. Group, "Embedded security ctf," NCC Group, (Accessed 05-27-2020). [Online]. Available: https://microcorruption.com/

[57] G. SSLab, "pwnable.kr," GaTech SSLab, (Accessed 05-27-2020). [Online]. Available: http://pwnable.kr/

[58] CyberTalents, "Cyber talents practice," CyberTalents, (Accessed 05-27-2020). [Online]. Available: https://cybertalents.com/challenges

[59] Google, "Xss game," Google, (Accessed 05-27-2020). [Online]. Available: https://xss-game.appspot.com/

[60] SDSLabs, "backdoor," SDSLabs, (Accessed 05-27-2020). [Online]. Available: https://backdoor.sdslabs.co/

[61] s4r, "crackmes," (Accessed 05-27-2020). [Online]. Available: https://crackmes.one/

[62] CTFLearn, "Ctflearn," CTFLearn, (Accessed 05-27-2020). [Online]. Available: https://ctflearn.com/

[63] HackerTest, "Hacker test," HackerTest, (Accessed 05-27-2020). [Online]. Available: http://www.hackertest.net/

[64] D. Keller, "Mr code's wild ride," (Accessed 05-27-2020). [Online]. Available: https://www.mrcodeswildride.com/hacking

[65] Netgarage, "Io wargame," Netgarage, (Accessed 05-27-2020). [Online]. Available: https://io.netgarage.org/

[66] P. Cobb, E. Yackel, and T. Wood, "A constructivist alternative to the representational view of mind in mathematics education," *Journal for research in mathematics education*, pp. 2–33, 1992.

[67] L. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1980. [Online]. Available: https://books.google.com/books?id=Irq913lEZ1QC

[68] L. Vygotsky, E. Hanfmann, G. Vakar, and A. Kozulin, *Thought and Language*, ser. Mit Press. MIT Press, 2012. [Online]. Available: https://books.google.com/books?id=B9HClB0P6d4C

[69] E. Seymour and N. M. Hewitt, *Talking About Leaving: Why Undergraduates Leave the Sciences*. Westview Press, 2000.

[70] P. Pusey, S. David Tobey, and R. Soule, "An argument for game balance: Improving student engagement by matching difficulty level with learner readiness," in *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*. San Diego, CA: USENIX Association, Aug. 2014. [Online]. Available: https://www.usenix.org/conference/3gse14/summit-program/presentation/pusey

[71] A. Naiakshina, A. Danilova, E. Gerlitz, E. von Zezschwitz, and M. Smith, ""if you want, i can store the encrypted password": A password-storage field study with freelance developers," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: ACM, 2019, pp. 140:1–140:12. [Online]. Available: http://doi.acm.org/10.1145/3290605.3300370

[72] A. Naiakshina, A. Danilova, C. Tiefenau, and M. Smith, "Deception task design in developer password studies: Exploring a student sample," in *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. Baltimore, MD: USENIX Association, 2018, pp. 297–313. [Online]. Available: https://www.usenix.org/conference/soups2018/presentation/naiakshina

[73] Y. Acar, M. Backes, S. Fahl, D. Kim, M. L. Mazurek, and C. Stransky, "You get where you're looking for: The impact of information sources on code security," in *Proceedings of the 37th IEEE Symposium on Security and Privacy*, ser. IEEE S&P, May 2016, pp. 289–305.

[74] Y. Acar, C. Stransky, D. Wermke, M. L. Mazurek, and S. Fahl, "Security developer studies with github users: exploring a convenience sample," in *Thirteenth Symposium on Usable Privacy and Security ({SOUPS} 2017)*, 2017, pp. 81–95.

[75] J. Novak, D. Gowin, C. U. Press, and J. Kahle, *Learning How to Learn*. Cambridge University Press, 1984. [Online]. Available: https://books.google.com/books?id=8jkBcSDQPXcC

[76] N. O. Lab, "Capture the flag | csaw," CSAW, (Accessed 05-27-2020). [Online]. Available: https://www.csaw.io/ctf

[77] G. A. Klein, *Sources of power: How people make decisions*. MIT press, 2017.

[78] A. L. Brown, "The development of memory: Knowing, knowing about knowing, and knowing how to know," ser. Advances in Child Development and Behavior, H. W. Reese, Ed. JAI, 1975, vol. 10, pp. 103 – 152. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0065240708600099

[79] W. G. Chase and H. A. Simon, "Perception in chess," *Cognitive psychology*, vol. 4, no. 1, pp. 55–81, 1973.

[80] F. Bartlett, "C.(1932). remembering," *Cambridge: CambridgeUniversityPress*, 1932.

[81] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive Science*, vol. 12, no. 2, pp. 257 – 285, 1988. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0364021388900237

[82] J. J. Van Merrienboer and J. Sweller, "Cognitive load theory and complex learning: Recent developments and future directions," *Educational psychology review*, vol. 17, no. 2, pp. 147–177, 2005.

[83] K. A. Ericsson and N. Charness, "Expert performance: Its structure and acquisition." *American psychologist*, vol. 49, no. 8, p. 725, 1994.

[84] E. Z. Rothkopf and M. Billington, "Goal-guided learning from text: inferring a descriptive processing model from inspection times and eye movements." *Journal of educational psychology*, vol. 71, no. 3, p. 310, 1979.

[85] D. R. Sadler, "Formative assessment: revisiting the territory," *Assessment in Education: Principles, Policy & Practice*, vol. 5, no. 1, pp. 77–84, 1998. [Online]. Available: https://doi.org/10.1080/0969595980050104

[86] A. S. Palincsar and A. L. Brown, "Reciprocal teaching of comprehension-monitoring activities," *Center for the Study of Reading Technical Report; no. 269*, 1983.

[87] M. Scardamalia, C. Bereiter, and R. Steinbach, "Teachability of reflective processes in written composition," *Cognitive Science*, vol. 8, no. 2, pp. 173 – 190, 1984. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0364021384800166

[88] A. H. Schoenfeld, "Problem solving in the mathematics curriculum. a report, recommendations, and an annotated bibliography. maa notes, number 1." 1983.

[89] P. J. Fadde and G. Klein, "Accelerating expertise using action learning activities," *Cognitive Technology*, vol. 17, no. 1, pp. 11–18, 2012.

[90] D. Hacker, J. Dunlosky, and A. Graesser, *Metacognition in Educational Theory and Practice*, ser. Educational Psychology Series. Taylor & Francis, 1998. [Online]. Available: https://books.google.com/books?id=EzWRAgAAQBAJ

[91] E. T. Pascarella and P. T. Terenzini, *How college affects students: Findings and insights from twenty years of research.* ERIC, 1991.

[92] A. M. O'Donnell and K. Alison, *Cognitive Perspectives on Peer Learning*, 1st ed. Routledge, 1999. [Online]. Available: https://doi.org/10.4324/9781410603715

[93] E. J. Whitt, M. I. Edison, E. T. Pascarella, A. Nora, and P. T. Terenzini, "Women's perceptions of a" chilly climate" and cognitive outcomes in college: Additional evidence." *Journal of College Student Development*, 1999.

[94] L. Watson, *How Minority Students Experience College: Implications for Planning and Policy*. Stylus, 2002. [Online]. Available: https://books.google.com/books?id=g6_7R-IX_E0C

[95] S. Hurtado, J. Milem, A. Clayton-Pedersen, and W. Allen, "Enacting diverse learning environments: Improving the climate for racial/ethnic diversity in higher education." *ASHE-ERIC Higher Education Report*, vol. 26, no. 8, 1999.

[96] Piazza, "Piazza - ask. answer. explore. whenever." Piazza, (Accessed 05-22-2020). [Online]. Available: https://piazza.com

[97] "Binary ninja cloud," Vector 35, 2020, (Accessed 06-03-2020). [Online]. Available: https://cloud.binary.ninja/

[98] L. J. Barker, K. Garvin-Doxas, and M. Jackson, "Defensive climate in the computer science classroom," in *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '02. New York, NY, USA: Association for Computing Machinery, 2002, pp. 43–47. [Online]. Available: https://doi.org/10.1145/563340.563354

[99] K. Garvin-Doxas and L. J. Barker, "Communication in computer science classrooms: Understanding defensive climates as a means of creating supportive behaviors," *J. Educ. Resour. Comput.*, vol. 4, no. 1, p. 2?es, Mar. 2004. [Online]. Available: https://doi.org/10.1145/1060071.1060073

[100] C. M. Lewis, K. Yasuhara, and R. E. Anderson, "Deciding to major in computer science: A grounded theory of students? self-assessment of ability," in *Proceedings of the Seventh International Workshop on Computing Education Research*, ser. ICER '11. New York, NY, USA: Association for Computing Machinery, 2011, pp. 3–10. [Online]. Available: https://doi.org/10.1145/2016911.2016915

[101] J. Margolis and A. Fisher, *Unlocking the clubhouse: Women in computing*. Cambridge, MA: MIT press, 2002.

[102] C. Crouch, A. P. Fagen, J. P. Callan, and E. Mazur, "Classroom demonstrations: Learning tools or entertainment?" *American Journal of Physics*, vol. 72, no. 6, pp. 835–838, 2004. [Online]. Available: https://doi.org/10.1119/1.1707018

[103] L. B. Nilson, "The graphic syllabus: A demonstration workshop on how to visually represent a course," in *Proceedings of the Professional and Organizational Development Network in Higher Education*, ser. POD '00, 2000.

[104] ——, "The graphic syllabus: Shedding a visual light on course organization," *To Improve the Academy*, vol. 20, no. 1, pp. 238–259, 2002. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-4822.2002.tb00585.x

[105] M. P. Robillard, "What makes apis hard to learn? answers from developers," *IEEE Software*, vol. 26, no. 6, pp. 27–34, 2009.

[106] D. L. Parnas and P. C. Clements, "A rational design process: How and why to fake it," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 2, pp. 251–257, 1986.

[107] T. C. Lethbridge, J. Singer, and A. Forward, "How software engineers use documentation: the state of the practice," *IEEE Software*, vol. 20, no. 6, pp. 35–39, 2003.

[108] StackOverflow, "Stackoverflow - where developers learn, share, and build careers," StackOverflow, (Accessed 06-05-2021). [Online]. Available: https://stackoverflow.com/

[109] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest Q&A site in the west," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: Association for Computing Machinery, 2011, pp. 2857–2866. [Online]. Available: https://doi.org/10.1145/1978942.1979366

[110] K. Yakdan, S. Dechand, E. Gerhards-Padilla, and M. Smith, "Helping johnny to analyze malware: A usability-optimized decompiler and malware analysis user study," *2016 IEEE Symposium on Security and Privacy (SP)*, vol. 00, pp. 158–177, 2016.

[111] E. L. Glassman, J. Scott, R. Singh, P. J. Guo, and R. C. Miller, "Overcode: Visualizing variation in student solutions to programming problems at scale," *ACM Trans. Comput.-Hum. Interact.*, vol. 22, no. 2, Mar. 2015. [Online]. Available: https://doi.org/10.1145/2699751

[112] A. Head, E. Glassman, G. Soares, R. Suzuki, L. Figueredo, L. D'Antoni, and B. Hartmann, "Writing reusable code feedback at scale with mixed-initiative program synthesis," in *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, ser. L@S '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 89–98. [Online]. Available: https://doi.org/10.1145/3051457.3051467

[113] R. Suzuki, G. Soares, A. Head, E. Glassman, R. Reis, M. Mongiovi, L. D'Antoni, and B. Hartmann, "Tracediff: Debugging unexpected code behavior using trace divergences," in *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2017, pp. 107–115.

[114] S. Bratus, "What hackers learn that the rest of us don't: Notes on hacker curriculum," *IEEE Security Privacy*, vol. 5, no. 4, pp. 72–75, 2007.

[115] G. Conti, T. Babbitt, and J. Nelson, "Hacking competitions and their untapped potential for security education," *IEEE Security Privacy*, vol. 9, no. 3, pp. 56–59, 2011.

[116] D. C. Rowe, B. M. Lunt, and J. J. Ekstrom, "The role of cyber-security in information technology education," in *Proceedings of the 2011 Conference on Information Technology Education*, ser. SIGITE ?11. New York, NY, USA: Association for Computing Machinery, 2011, p. 113?122. [Online]. Available: https://doi.org/10.1145/2047594.2047628

[117] V.-V. Patriciu and A. C. Furtuna, "Guide for designing cyber security exercises," in *Proceedings of the 8th WSEAS International Conference on E-Activities and Information Security and Privacy*, ser. E-ACTIVITIES/ISP 09. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2009, pp. 172–177.

[118] J. Kick, "Cyber exercise playbook," MITRE CORP BEDFORD MA, Tech. Rep., 2014.

[119] P. Pusey, M. Gondree, and Z. Peterson, "The outcomes of cybersecurity competitions and implications for underrepresented populations," *IEEE Security Privacy*, vol. 14, no. 6, pp. 90–95, 2016.

[120] A. Ruef, M. Hicks, J. Parker, D. Levin, M. L. Mazurek, and P. Mardziel, "Build it, break it, fix it: Contesting secure development," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 690–703. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978382

[121] T. Reed, K. Nauer, and A. Silva, "Instrumenting competition-based exercises to evaluate cyber defender situation awareness," in *Foundations of Augmented Cognition*, D. D. Schmorrow and C. M. Fidopiastis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 80–89.

[122] P. Chapman, J. Burket, and D. Brumley, "Picoctf: A game-based computer security competition for high school students," in *Proc. of the 1st USENIX Summit on Gaming, Games, and Gamification in Security Education*, ser. 3GSE '14. San Diego, CA: USENIX Association, 2014. [Online]. Available: https://www.usenix.org/conference/3gse14/summit-program/presentation/chapman

[123] K. Maennel, R. Ottis, and O. Maennel, "Improving and measuring learning effectiveness at cyber defense exercises," in *Secure IT Systems*, H. Lipmaa, A. Mitrokotsa, and R. Matulevičius, Eds. Cham: Springer International Publishing, 2017, pp. 123–138.

[124] S. Karagiannis, E. Maragkos-Belmpas, and E. Magkos, "An analysis and evaluation of open source capture the flag platforms as cybersecurity e-learning tools," in *Information Security Education. Information Security in Action*, L. Drevin, S. Von Solms, and M. Theocharidou, Eds. Cham: Springer International Publishing, 2020, pp. 61–77.

[125] M. Whitney, H. Lipford-Richter, B. Chu, and J. Zhu, "Embedding secure coding instruction into the ide: A field study in an advanced cs course," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE ?15. New York, NY, USA: Association for Computing Machinery, 2015, p. 60?65. [Online]. Available: https://doi.org/10.1145/2676723.2677280

[126] C. Weir, L. Blair, I. Becker, A. Sasse, and J. Noble, "Light-touch interventions to improve software development security," in *2018 IEEE Cybersecurity Development (SecDev)*, 2018, pp. 85–93.

[127] A. Poller, L. Kocksch, S. Türpe, F. A. Epp, and K. Kinder-Kurlanda, "Can security become a routine? a study of organizational change in an agile software development group," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ser. CSCW '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 2489?2503. [Online]. Available: https://doi.org/10.1145/2998181.2998191

[128] A. T. Sherman, D. DeLatte, M. Neary, L. Oliva, D. Phatak, T. Scheponik, G. L. Herman, and J. Thompson, "Cybersecurity: Exploring core concepts through six scenarios," *Cryptologia*, vol. 42, no. 4, pp. 337–377, 2018. [Online]. Available: https://doi.org/10.1080/01611194.2017.1362063

[129] J. Blasco and E. A. Quaglia, "Infosec cinema: Using films for information security teaching," in *2018 USENIX Workshop on Advances in Security Education (ASE 18)*. Baltimore, MD: USENIX Association, 2018. [Online]. Available: https://www.usenix.org/conference/ase18/presentation/blasco

[130] P. Deshpande, C. B. Lee, and I. Ahmed, "Evaluation of peer instruction for cybersecurity education," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 720–725. [Online]. Available: https://doi.org/10.1145/3287324.3287403

[131] W. E. Johnson, A. Luzader, I. Ahmed, V. Roussev, G. G. R. III, and C. B. Lee, "Development of peer instruction questions for cybersecurity education," in *2016 USENIX Workshop on Advances in Security Education (ASE 16)*. Austin, TX: USENIX Association, 2016. [Online]. Available: https://www.usenix.org/conference/ase16/workshop-program/presentation/johnson

[132] T. Denning, A. Shostack, and T. Kohno, "Practical lessons from creating the control-alt-hack card game and research challenges for games in education and research," in *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*. San Diego, CA: USENIX Association, Aug. 2014. [Online]. Available: https://www.usenix.org/conference/3gse14/summit-program/presentation/denning

[133] S. Frey, A. Rashid, P. Anthonysamy, M. Pinto-Albuquerque, and S. A. Naqvi, "The good, the bad and the ugly: A study of security decisions in a cyber-physical systems game," *IEEE Transactions on Software Engineering*, vol. 45, no. 5, pp. 521–536, 2019.

APPENDIX A
INTERVIEW PROTOCOL

Over the past few months, we have been looking at several online capture-the-flag competitions, wargames, and other security exercises with an educational focus. For each exercise, we completed several challenges reviewing the content and the way in which this information was presented to students.

Our goal was to see how each exercise is organized, focusing on features commonly recommended in the learning science and education literature. This includes features related to connecting to the learner's prior knowledge, the organization of knowledge, providing active practice and feedback, encouraging metacognitive learning, and establishing a supportive environment.

In this interview, I want to talk about each of these categories, the specific dimensions of each, and how we viewed these as being applied in your exercise. We have a few goals with this interview. The first is to make sure we have a clear picture of your exercise. You know it much better than we ever will, so we want to tap into your knowledge to make sure we're not missing something. Second, we want to understand more about why your exercise is organized the way it is. The dimensions we're looking at are recommendations, not requirements. As you'll see, they are many good things and exercise might have. We would expect an exercise to have some, but not all. So, in this interview, we want to get at why you decided to include some, but not others.

A. General Organizational Questions

Before we begin going through our review, I have a few general questions about your organization to help us better understand the context in which the exercise was developed.

1) What lead you to create your exercise? That is, what was your main motivation in providing this educational platform?
2) Who is your target audience(s)?

B. Analysis Review

Moving on to our review of your exercise, now I'll go through each core pedagogical principle and its dimensions. For each, I will give you our definition and our decision and then get your response.

For each core principle and related dimension, we provided the interviewee with the textual definitions given throughout Section III

For each dimension coded No or Partial, we explained the reasoning behind our decision and asked the following question progression:

1) Do you agree?
   a) *If no:* Why not?
   b) *If yes:* Do you agree that this pedagogical dimension would be helpful to students if implemented?
      i) *If no:* Why not?
      ii) *If yes:* Why did you choose not to implement it?
         A) *If they said it was too difficult:* What are the challenges?

APPENDIX B
CODEBOOK

In this appendix, we provide the codebook used to categorize organizer decisions for not fully implementing reviewed pedagogical principles.

- **Does not fit our goals**: The organizer considered implementing the dimension, but chose not to because it did not fit within the structure and intended goals of the exercise.
- **Challenging**: The organizers considered implementing the dimension, but decided that implementation would be too difficult.
- **Not considered**: The organizers did not consider the dimension when designing the exercise.

APPENDIX C
ADDITIONAL DIMENSIONS

In Table II, we present the results of our pedagogical review that did not show much variety between exercises. These dimensions were not included in Table I due to space considerations.

	Content						**Active Practice**	**Metacognition**		Challenges reviewed
	Reverse Engineering	Cryptography	Web Vulnerabilities	Binary Exploitation	Forensics and Networking		Exploit Code		How to use	
Synchronous										
gCTF† [36]	●	●	●	●	●		●		●	5
iCTF† [37]	●	●	●	●	●		●		●	7
Infosec Institute [38]	○	●	●	○	●		◐		●	15
picoCTF† [39]	●	●	●	●	●		●		●	10
CSAW365 [40]	●	●	●	●	●		●		●	14
HackEDU [41]	●	●	●	●	○		●		●	3*
Pwnadventure† [42]	●	●	◐	●	○		●		●	5
PACTF [43]	●	●	○	○	●		●		●	15
Angstrom† [44]	●	●	●	●	●		●		●	7
HXP CTF [45]	●	●	●	●	●		●		●	5
BIBIFI† [46]	●	●	○	●	●		●		●	N/A
Pwn College† [47]	●	○	○	●	●		●		●	10
GirlsGo CyberStart† [48]	●	●	●	○	●		●		●	14
Asynchronous										
HackTheBox† [49]	●	●	●	●	●		●		●	15
HackthisSite [50]	●	●	●	○	●		●		●	11
OverTheWire [51]	●	●	●	●	●		●		●	15
Root-me.org† [52]	●	●	●	●	●		●		●	11
Vulnhub† [53]	●	●	●	●	●		●		●	5
Hacker101 [2]	●	●	●	●	●		●		●	9
Hellbound Hackers [54]	●	●	●	●	●		●		●	17
Smash the Stack† [55]	●	●	●	●	○		●		●	10
Microcorruption [56]	●	○	○	●	○		●		●	5
Pwnable [57]	●	●	○	●	○		●		●	8
Cyber Talents [58]	●	●	●	●	●		●		●	7
XSS-Game† [59]	●	○	●	○	○		●		●	6
Backdoor [60]	●	●	●	●	●		●		●	7
Crackmes.one† [61]	●	●	○	●	○		●		●	5
CTFlearn [62]	●	●	●	●	●		●		●	6
HackerTest [63]	○	○	●	○	○		●		●	10
Mr. Code† [64]	●	●	●	○	●		●		●	10
IO Wargame [65]	●	○	○	●	○		●		●	7

† An organizer from this exercise was interviewed or responded via email to our review.
* HackEDU only offered three publicly available challenges.

TABLE II: Dimensions of our pedagogical analysis not covered in Table I. Each column indicates whether an exercise implemented the pedagogical dimension fully (●), partially (◐), or not at all (○). Additionally, the final column gives the number of challenges reviewed when evaluating each exercise.