Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #48224

Hacking the System: A Peer-Led Cybersecurity Course for Early-Career University Students

Mr. Ian Hong Phan, University of California, Santa Cruz

Ian Phan is a recent graduate in Cognitive Science from the University of California, Santa Cruz. He has been involved in the Baskin School of Engineering's First-Year Design program for two years, serving as a student instructor and coach for teaching teams developing new courses. His work focuses on advancing collaborative STEM education, promoting interdisciplinary collaboration, and reducing barriers to success in engineering fields. Beyond the classroom, he mentors high school robotics teams and conducts research on participatory programs that expand access to undergraduate research opportunities—further reinforcing his commitment to inclusive and accessible education. Through his work, he aims to foster curiosity, collaboration, and confidence among all students, preparing them to tackle complex, real-world challenges.

Iakov Taranenko, University of California, Santa Cruz

Iakov Taranenko is a fourth year computer networking undergraduate at the University of California, Santa Cruz. For two years, he served as a student instructor for a first year design program specializing in cybersecurity, helping underclassmen explore the cybersecurity field. Extremely passionate about cybersecurity, he brings valuable knowledge from his employment as a systems exploitation engineer within government roles and his participation in graduate-level cybersecurity research. This experience extends beyond his academic work into student leadership, where as president of the university's cybersecurity student organization, he organizes hands-on workshops, mentors team members for national cybersecurity competitions, and creates opportunities for students to test their hacking skills against realistic simulated systems. Under his leadership, the club has significantly increased membership and regularly places among the top teams in collegiate cybersecurity challenges across the country.

Dr. Tela Favaloro, University of California, Santa Cruz

Tela Favaloro is an associate teaching professor for the Baskin School of Engineering at UCSC where she works to establish holistic interdisciplinary programming centered in experiential learning. Her Ph.D is in Electrical Engineering with emphasis in the design and fabrication of laboratory apparatus and techniques for electro-thermal characterization of sustainable power systems as well as the design of learner-centered experiential curriculum. She is currently working to develop an inclusion-centered first-year engineering program in hands on design and problem-based learning to better support students as they enter the engineering fields.

Hacking the System: A Peer-Led Cybersecurity Course for Early-Career University Students

Introduction: A Pressing Need

In a world where technology is increasingly prevalent, the widespread adoption of digital systems has fundamentally reshaped how we live, work, and connect, increasing the need to keep these systems - and our personal data that traverse these systems - secure. However, growth in the cybersecurity sector has not matched the complexity of the threat landscape; new vulnerabilities introduced through artificial intelligence and the Internet of Things broaden an already wide attack surface and demand specialized skills to address. Organizations worldwide are struggling to keep pace with these escalating threats, straining the resources and capabilities of their existing cybersecurity teams and further underscoring the need for a skilled workforce. Yet, the cybersecurity industry is currently facing a significant skills gap; in 2024 there was found to be an estimated global shortage of 4.8 million cybersecurity professionals. Workforce growth has plateaued at around 5.5 million globally, while the skill gap widened by 19% compared to the previous year [1]. In the United States alone, the supply of cybersecurity professionals met only 83% of employer demand, leaving over 225,000 positions unfilled as of June 2024 [2, 3, 4].

According to recent industry reports, professionals with the following technical and professional skills are needed to fill this shortage in the cybersecurity workforce [1, 5]:

- **Technical skills**: general hands-on expertise in Linux systems and cloud platforms; specialized experience in: network security (2nd largest skill gap), threat detection, malware analysis, cryptography, data privacy, forensics, and reverse engineering.
- **Professional skills**: cybersecurity leadership (the largest skill gap), adaptability, teamwork and collaborative skills, effective communication, as well as problem-solving and critical thinking. These are all skills associated with the development of a *hacker mindset*, a flexible mindset which approaches problems with curiosity, creativity, and a determination to explore unconventional solutions.

Several factors are credited as contributing to the industry's ever-growing cybersecurity skill gap. Chiefly, there is an insufficient number of educational programs to meet modern cybersecurity demands. Many academic institutions struggle with aligning their curricula with the fast-evolving needs of the cybersecurity industry. Collaboration between academic institutions and the wider industry remains limited, divorcing faculty expertise and leaving graduates underprepared for roles in emerging areas such as cloud computing, AI-driven security, and threat intelligence [5]. Programs that do exist often focus heavily on foundational theory while neglecting hands-on training. Furthermore, the shortage of *accessible*, advanced training opportunities for skill development, such as certifications or boot camps, hampers the ability to meet workforce needs [1].

Even with these barriers in education, the industry continues to utilize traditional hiring practices that prioritize educational credentials over demonstrable skills, effectively limiting the talent pool. Paradoxically, industry has begun turning to other methods to gauge potential new hires' proficiency in the sought-after skills and knowledge: paid learning platforms with certifications and Capture-the-Flag (CTF) competition experience [5]. As mentioned above, the former is very

pricey and out of reach for most new learners. Consequently, CTFs are beginning to serve as a substitute for academic achievement in these fields.

CTFs ask participants, either on their own or in a team, to complete a number of CTF-style challenges in different skill areas to gain points over a span of time which may range from one night to months in duration. There are many popular platforms of CTFs, some attracting over 27,000 participants with each offering [6]. CTF challenges are often bite-sized, each focused on showcasing a specific skill or a subset of skills in pursuit of tokens of completion, referred to as "flags." The competition platforms themselves are modular; challenges are organized around a specific theme then may progress in targeted complexity towards advanced competency or, alternatively, demand diverse and broad skill sets to succeed. Pedagogically, CTFs competitions are a method of gamified learning. The above "zero to hero" progression used by many CTFs translates to an incremental approach to cybersecurity education that effectively scaffolds skill development. New participants begin with easier challenges that require little to no knowledge or experience and then progress to harder challenges as they master concepts.

The University Context

While the cybersecurity industry continues to grow and evolve at a rapid pace, our universities, and by extension our students, continue to trail behind. Academic institutions are generally unable to offer courses with content that aptly prepares and evaluates students' ability to apply their skills to the challenges faced in industry. To understand why, it is important to first recognize the current state of cybersecurity education in our universities. Advanced cybersecurity education is a newfangled topic; cybersecurity has just recently split into its own discipline where previously it would be under university departments such as computer science or computer engineering. Unlike industry, there is considerable latency built into the academic system due to lengthy course approvals, limited infrastructure build-up, and mismatch of skill offered by faculty. Large class sizes found in many universities do not easily accommodate novel, resource-intensive classes that require significant support and maintenance, as well as the flexibility to reframe class learning outcomes quickly to match industry needs.

Given that cybersecurity is a new and rapidly changing field, a plug-and-play platform and accompanying infrastructure to teach cybersecurity does not exist, unlike, for example, the ubiquitous electronics lab bench used for teaching circuit design. As a result, there are few true *vetted* models of teaching cybersecurity, especially for supporting beginner-level students who are just starting to interact with the field. Even seemingly simple tasks, such as planning in-class activities, can be difficult to manage without extensive forethought and secure infrastructure development; hacking can get quite messy at times and could potentially pose security risks for the university. Both faculty and staff must coordinate to find holes in the infrastructure and stop students from doing anything potentially dangerous. Without proper preparation and knowledge of systems, potential oversights could have major consequences, such as a student inadvertently scanning the university-wide network instead of a dedicated isolated networking sandbox designed for such work. Universities must develop not only their own program but also the associated infrastructure - often servers and proprietary equipment - from scratch, demanding significant resources and funding just to get started. Thus, sustaining *applied* cybersecurity programs requires significant investment of both time and money for institutions to host and maintain equipment, let alone quickly adapt to new technology and threats in the field.

We postulate these difficulties limit the availability of experiential learning within university cybersecurity programs, restricting classes to graduate students with smaller cohorts and leaving early learners without a pipeline to get interested and start in the field. Limited accessibility is a setback to any university program but especially cybersecurity as every student brings diverse perspectives - a benefit to flexible and innovative thinking that defines the *hacker mindset*, but only if we are able to support different learning styles through considered instructional design.

Unfortunately, all these challenges likely culminate in what could be the biggest issue of all: interested students are inhibited from joining the field. Lack of perceived demand from the student population leads to fewer courses being offered in the area. However, it may be the universities themselves that are fueling student interests, or lack thereof. Students who may have been motivated to pursue cybersecurity-related fields are often a minority in university programs (e.g. Computer Science) and feel compelled to switch to areas better served by the university. This leads to fewer students showing long-term interest in the field and, ultimately, fewer classes being offered. Moreover, the difficulty of implementing experiential learning in cybersecurity further alienates students who get involved due to its cool factor, those who learn better by doing (which is most students), or those that are motivated by the real-life applications of their work and don't want to wade through long chains of prerequisites to get there. Despite this, the demand *is there* and students such as us are leading the way; cybersecurity-related clubs are popping up on campuses across the US in an attempt to bring cybersecurity learning to students. While cybersecurity is something that could be run independently by students, it perhaps should not be without university support due to the risks alluded to above.

As undergraduate students ourselves, our solution to these issues was to take control of our engineering curriculum to formally create a modular course and its associated infrastructure that guides early-career undergraduate students through introductory and intermediate cybersecurity. Our course, *Introduction to Hacking Competitions,* is based in active and experiential learning through home-built Capture the Flag (CTFs) challenges, targeting the development of both technical and professional skills in a scaffolded peer-learning environment. In this proceeding, we present the intentional design of the 10-week introductory course that does not require any prerequisite knowledge, describe its modular course flow, then depict the technical infrastructure we designed to host the CTF challenges in the different topic areas. Finally, we discuss the results from the learner perspective drawn from data gathered over two, quarter-long offerings through externally administered surveys, interviews, and observations. We find that the course achieves our ultimate goal as student-instructors: to support learners in developing not only technical proficiency but also the *hacker mindset* to become adaptable and capable thinkers in the cybersecurity field.

A Modular and Accessible Cybersecurity Course

Introduction to Hacking Competitions is a new class that challenges a system that unfortunately limits early engineering experiences to theory, physics, math, and introductory programming. Our university, a public, R1, MSI institution, is part of a consortium of 10 universities across the state that together host ~65 formal classes in areas related to cybersecurity (with 1 certificate program). However, only 19 of these are available to undergraduate students, all of which are restricted to upper-division enrollment (note, some advanced undergraduates are able to enroll in

the graduate level courses). Our university is much the same, with one graduate-level course (Advanced Computer Security) and one upper-division undergraduate-level class related to Cybersecurity (Introduction to Computer Security) that is restricted to Engineering majors and has a sequential string of three prerequisite courses. While there may be more classes on the horizon, all of these are spearheaded by a single faculty member in response to the growing demand for cybersecurity professionals.

Intro to Hacking Competitions is a 3-unit (9 work-hours per week) course that satisfies a General Education requirement and is supported by the First Year Design program hosted by our School of Engineering. This program brings professional engineering practice to early-career students through a series of Design-Build courses that cover concepts and experiences often reserved for advanced- or senior-level students in their capstone courses [7, 8]. Courses within the First Year Design program are developed and taught by students, for students; they span a plethora of Design-Build topics designed to bolster skills in areas closer to students' interests, with *Intro to Hacking Competitions* being one of many fun, interesting, and interactive ways to gain exposure to these topics. By employing students as the teachers (under faculty guidance) in a classroom limited to 20 learners, hands-on and experiential learning becomes more accessible and interactive; many learners build relationships that continue outside of the classroom [7, 8, 9]. Often student-teachers join the program not only to make sure important knowledge and skills are accessible to early learners, but also to take control of the engineering curriculum and modernize its offerings.

Motivations of a Student Teacher

The central idea behind undergraduate student-led courses as part of the First-Year Design program is to increase diversity and retention within engineering by providing opportunities for engineering practice in an active encouraging environment, one that differs from common experiences of first-year learners in traditional lecture settings [7, 8, 9]. The following statements detail our motivations as undergraduate student-teachers to not only participate in the First Year Design program but to play a role in advancing engineering education as a whole:

- **Ian Phan** (Cognitive Science, 3rd year) was first introduced to the world of cybersecurity and CTFs in senior year of high school and into his first year of college. As a student outreach officer for their university's cybersecurity club, he felt that developing a class and teaching beginners interested in the field would be a meaningful way to make the field more accessible and inspire others to take their first steps into the wide world of cybersecurity. As a student-teacher, his overall objective is to create more equitable learning spaces for new students to the field, fostering learner curiosity.

- **Iakov Taranenko** (Network & Digital Technology, 3rd year). Participant in cybersecurity competitions (5+ years), employed as Systems Exploitation Engineer (3 years), general focus on using gamified cybersecurity platforms as learning tools. Founded and currently runs the university's cybersecurity club, working to make cybersecurity more welcoming to students in traditional academic pathways by teaching the skills needed for participation as well as the ethical perspective. By teaching in a formal setting, he aims to share the excitement of hands-on learning in cybersecurity and inspire more students to explore the field.

We designed *Hacking Competitions* to foster intermediate-level proficiency in technical and professional skills that align with the direction of the cybersecurity industry, while being

accessible to learners without any prerequisite background. The course is structured around custom-designed CTF challenges that we develop and host on our dedicated, home-built platform. Over the course of the quarter, students complete a variety of challenges in different cybersecurity topics, both in class and at home, as individuals and in groups, to experientially build their skill in preparation for a culminating final experience. Over this two-week final project challenge, student-teams are free to flex their skills in our own CTF competition to achieve points towards the completion goal. This challenge-driven yet collaborative learning environment keeps learners engaged throughout the quarter while removing the competitive framework that often drives hacking competitions. As depicted in **Table 1** below, the course develops a strong foundation in Linux systems in addition to specialized hacking skills, providing learners with the tools to respond to the needs of the current cybersecurity landscape. Furthermore, *Hacking Competitions* equips students with the *soft* engineering process skills that build the *hacker mindset*, preparing students for professional engineering practice.

***Table 1**: Hacking-specific (red) & general engineering process skills (green) as a table of learning outcomes for Intro to Hacking Competitions. Italicized items are required in all First Year Design classes.*

Technical Skills	**Professional Engineering Skills**
Computer Literacy in advanced tasks such as command line/terminal usage	*Professional documentation practices through the use of an engineering notebook*
Understand and know when to apply various Linux commands	*Ability to evaluate properties and features to meet functional requirements under known constraints*
Analyze and interpret different forms of digital forensics data (metadata, network/protocol data)	*Familiarity with the engineering design cycle, iterative design, and version control as demonstrated through successive prototypes*
Understand and exploit web vulnerabilities in sites and applications	*Collaborate effectively on a cross-disciplinary team to co-develop solutions under constraints*
Examine and make sense of various unfamiliar programming code formats.	Plan and allocate resources efficiently to ensure timely completion of engineering tasks
Bonus: Interpret various forms of real-world attacks /vulnerabilities and their implications (CVEs/CWEs)	Demonstrate advanced research skills and effective resource gathering
Bonus: Demonstrate basic skills in WiFi Hacking	Utilize analytical thinking to dynamically solve problems
Bonus: Apply knowledge of SQL databases and different forms of database injection/exploitation techniques	Demonstrate effective disciplinary communication to diverse audiences through technical writeups & presentations
Bonus: Understand fundamentals of malware basics and reversing	Exhibit persistence and flexibility when navigating complex, multilayered challenges

A Modular Design: the high-level course flow

One challenge in experientially teaching advanced or complex topics such as cybersecurity and hacking is that they potentially can be, and often are, out of reach for beginners in the field. To address this, our course utilizes active learning framed by high structure, which has been shown to close achievement gaps among learners [8]. Furthermore, we employ a method of scaffolding

in our course content designed to gradually bring learning and concepts into students' zone of proximal development (refer to **Figure 1** below). Students slowly build a practiced foundation by incrementally moving through modular topics via associated CTF challenges, transitioning from "I can't do this" mindset to "I can do this with help" and ultimately "I CAN do this" by the end of the quarter. Over the ten-week quarter, every few weeks of *Intro to Hacking Competitions* focuses on a specific set of skills in cybersecurity and hacking while continuously building competency with the Kali distribution of the Linux operating system.

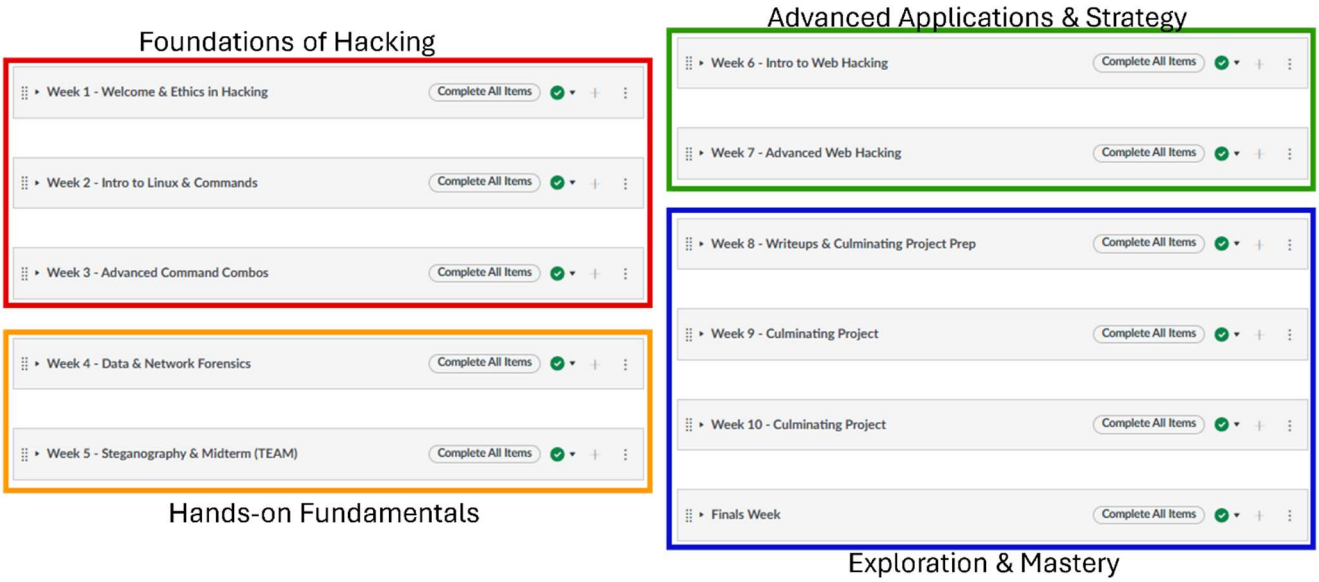

Figure 1*: Ten-Week Course Flow. Intro to Hacking Competitions is split into 4 stages, each focusing on a different stage of skill development. Through completion and mastery of each stage of the course, students progress further towards autonomy, eventually reaching the final stage of the course where they have the capability to start exploring and mastering cybersecurity topics / challenges on their own.*

Introduction to Hacking Competition's course design can be broken down into four stages of skill complexity. For the first half of the quarter, students complete the first two stages, 1) *Foundations of Hacking* and 2) *Hands-On Fundamentals*, culminating in a midterm to assess and cement knowledge in these foundational topics and ensure that learners are prepared to build on this work. *Foundations of Hacking* focuses on ethical hacking, hacking culture, Linux, and command-line basics, while *Hands-on Fundamentals* expands on the *Foundations,* teaching external and custom applications with weekly CTF challenges in data & network forensics and steganography. For the take-home midterm, students are paired up - where they will remain for the remainder of the quarter - and are tasked with completing three key CTF challenges in basic & advanced commands and data & network forensics.

The latter half of the course moves on to advanced topics with increased learner independence: 3) *Advanced Applications & Strategies* leading into the final stage, 4) *Exploration & Mastery*, a culminating course experience as a multi-week mock CTF competition. Learners continue to build upon fundamentals while exploring new topics as a team, such as various methods of web hacking and exploitation. In addition, students select bonus topics to be covered, which have included SQL vulnerabilities, malware, real-world attacks, and WiFi hacking.

The real beauty and magic of our modular stages of skill development is that later topics can easily be swapped; the fundamentals set the foundation for many possibilities in advanced topics.

As the landscape of cybersecurity evolves at a rapid pace, concepts that we are teaching now may well be outdated and irrelevant in the near future. Beyond the first stage in our course, the technical topics covered during the course may easily and incrementally be adapted by modifying or designing new CTF challenges while preserving the gradual development of soft skills that formulate the *hacker mindset*. This adaptability ensures that the course remains valuable, equipping students with relevant technical proficiency on a platform that is more easily maintained by university staff. Overall, the modular design of this course prepares students to tackle emerging challenges and technologies, ensuring they remain lifelong learners and effective problem solvers in the field.

A Familiar Tempo for Learning: the high-structure breakdown of curricular stages

In each week of our course, we follow a three-step cycle of skill development designed to increment the level of challenge while building a familiar tempo for students to complete their work. Students begin the week with a low-stakes introduction to the content, then build on these topics by collaboratively completing low-stakes CTF challenges as in-class activities. Finally, learners work independently at home on challenges designed to synthesize their learning for that week. Figure 2 displays an example structure during *Hands On Fundamentals* (Weeks 4 and 5).

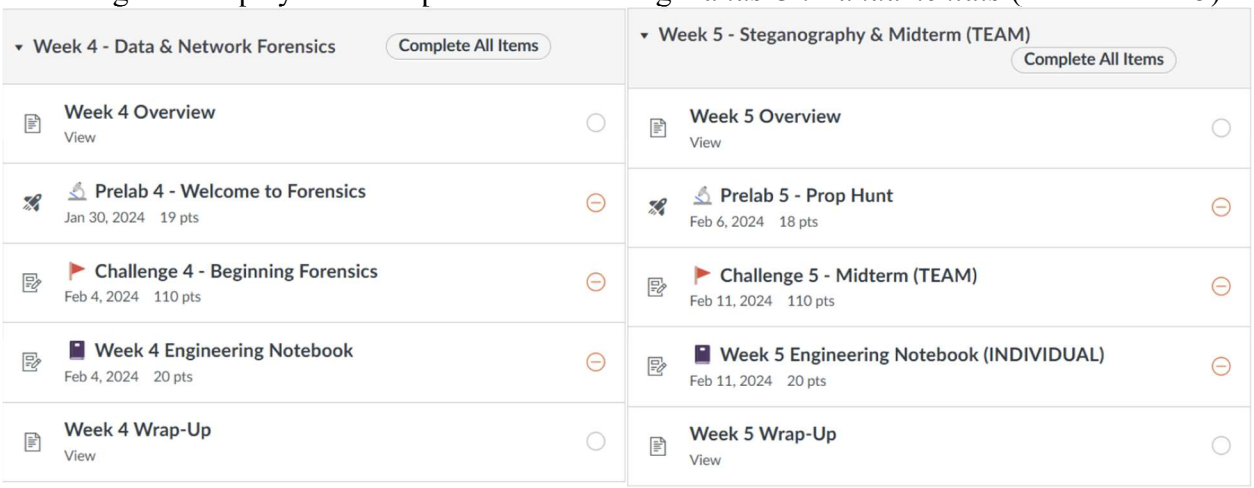

Figure 2*: Overview of student-facing content of Weeks 4 and 5 of the course, as presented in the Course Learning Management System. In these weeks, students are introduced to the idea of forensics in digital form (such as in files, photos, videos, and networks), as well as means of data concealment, including steganography. Each of these weeks follows the 3-step process of prelab → in-class active learning → lab challenge homework with associated professional documentation.*

Prelab: Each week begins with a prelab, where learners are assigned content to read and/or watch that introduces the topics of the week, followed by a short assessment to ensure completion and reinforce learning. By beginning each week with a prelab, we *flip* the classroom, encouraging students to proactively engage with the material by exploring practical applications and potential implementations before they are formally introduced in class. Prelab quizzes ensure students engage with foundational material before delving into these new topics, helping to bridge gaps in understanding while positioning learners to build on these topics through more effective in-class participation.

In-class Active Learning: Each week is punctuated by two 95 minute in-person sessions where students learn, explore, and work hands-on with custom-developed CTF challenges. This way, they collaboratively build and cement skills and knowledge while developing proficiency in the week's topic. This may mean practicing and experimenting with Linux commands/programs,

delving into different types of data, or poking around the backend of a website looking for vulnerabilities to exploit (example shown in Figure 3a). The teaching team designs the in-class CTF challenges as a quick way to practice technical skills in a supported manner, building confidence in both independent and collaborative analytical thinking and problem solving, which are key competencies not just in cybersecurity, but in the field of engineering as a whole.

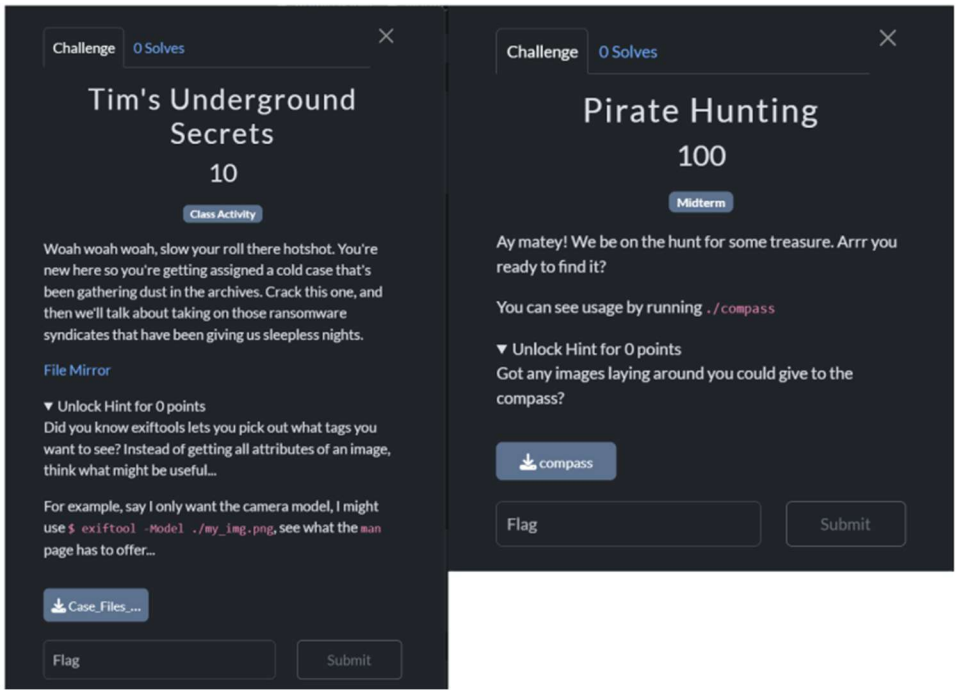

Figure 3: 3a (LHS) *depicts an example of an in-class challenge given during Week 4, where students learn about the role of metadata in digital forensics, which may include a device's make/model, time, and location. Students use this understanding to analyze the metadata of a series of photos to piece together a map that shows the locations and paths of alleged individuals suspected of committing corporate espionage against an undisclosed Canadian coffeehouse/restaurant chain.* ***3b (RHS)*** *gives an example homework challenge from the midterm assignment in Week 5. During Week 4, students learn that metadata can not only be read but also altered. The latter strategy is not explicitly practiced during class but is needed to complete this challenge. As we encourage an exploratory mindset, the hint prompts students to see what happens if an image is run via the compass program.*

Lab Challenge Homework: The week culminates with three take-home CTF challenges, completed individually during Stages 1 and 2, or as a pair during Stage 3, giving learners a chance to independently apply what they have learned in a real-world context. Homework challenges are designed and curated in a way that supports students' continued learning while demonstrating proficiency in the week's topics. Specific tools or concepts that are required to solve the challenges are not explicitly conveyed in the assignment, requiring students to utilize and build upon their analytical thinking, research processes, and creative problem-solving skills in order to solve the challenge while effectively stretching the boundaries of the students' zone of proximal development. An example of a homework challenge is given in Figure 3b.

Professional Documentation: In addition to cybersecurity-specific assignments, *Intro to Hacking Competitions* develops learners' proficiency in producing technical documentation, a crucial part of any engineer's skill set. Documentation in our course comes through different mediums: an individual student's engineering notebook and team technical challenge writeups. These documentation methods are an important part of one's journey not just to be a proficient

hacker but an engineer as well - as such, it serves an important role as part of the students' arsenal of tools. Everything related to the course goes into the notebook, including, but not limited to, lecture/activity notes, prelab notes, team discussions, challenge planning, thought processes, and mistakes made during challenge completion. Notebook content on CTF challenges is then used as source material for the ***technical writeups*** to communicate strategy and learning. Writeups in the cyber and hacking space are an essential form of technical communication, serving as a way to document methodologies, share insights, and demonstrate problem solving processes and technical proficiency. Writeups are used ubiquitously in CTF competitions as part of determining the winner as well as in professional contexts to convey how challenges or tasks are approached, the tools and techniques utilized, and the reasoning behind solutions. Writeups are an integral component of our course; we begin assigning technical writeups with the team-based midterm project, however, engineering notebook prompts and processes leading up to the midterm ease learners into creating this more official documentation. In this way, students practice industry-relevant documentation skills while reinforcing and elucidating their understanding of concepts covered in class.

Culminating Final Project & Presentations: Students wrap up their journey into hacking with a pair-based culminating project as a mock Capture the Flag (CTF) competition, which consists of a bank of ~40 challenges categorized by the technical topics covered in the course, including the student-selected bonus content. Student-teams pick and choose which challenges they complete to meet the completion point-threshold, as long as they meet minimum requirements of three challenges in each of the main course topic categories:

- General Linux Skills: e.g. reverse command actions to uncover a secret message in a text file.
- Forensics & Steganography: e.g. recover a lost password through exploiting a weakness of a network transfer protocol.
- Web Hacking: e.g. exploit a XSS vulnerability to retrieve a user's 'cookie.'
- Exploratory Bonus Topics: e.g. use SQL injections to dump a database of sensitive data.

In these CTF challenges, students are tasked with first analyzing prompts, utilizing critical thinking skills to link prompts to technical concepts, then leveraging both their technical knowledge and strategic thinking skills to complete challenges across the various domains covered in the course. Each challenge is worth ~100 points; student-teams are required to earn 1000 points per person over a two-week period to successfully complete the final challenge. The final CTF competition allows students to showcase their achievement of the course's learning outcomes, emphasizing: their ability to approach complex cybersecurity challenges, apply foundational tools and techniques, as well as adapt to dynamic problem-solving scenarios. Figure 4 depicts the final CTF challenge bank during the Fall 2024 offering. The class wraps up during the final exam period with a DEF-CON (hacker conference) style presentation, designed to celebrate the students' accomplishments over the quarter in a light, party-like atmosphere. During this period, student-teams present a 10-minute synopsis of their solution to one challenge, assigned from their library of completed challenges. This interactive format encourages sharing diverse strategies, intense discussions, and collaborative problem-solving, fostering a dynamic learning environment that mirrors real-world practices.

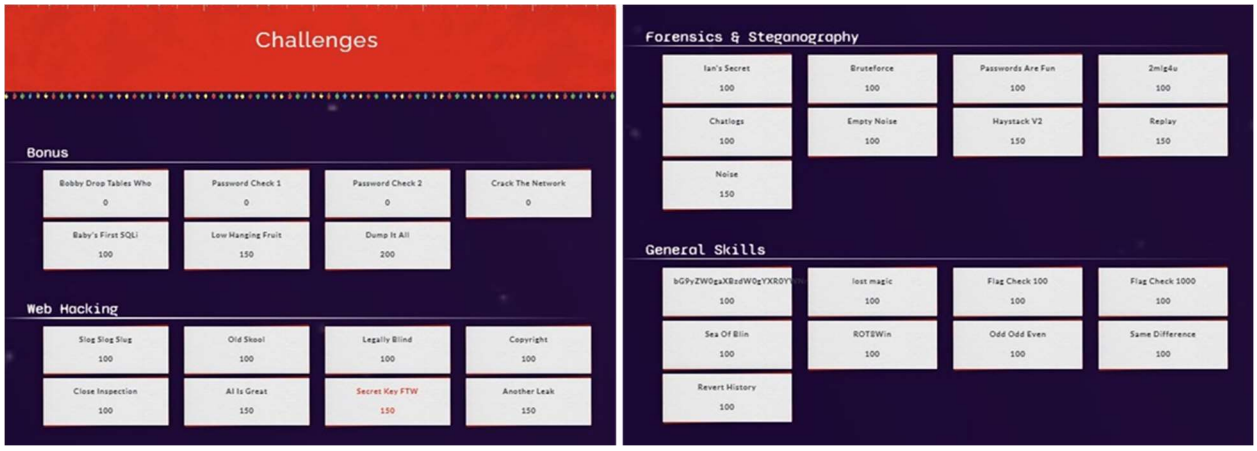

Figure 4*: Student-facing view of final challenge bank for Fall 2024 offering. 30+ home-built challenges are available to students to select from, spread across the 4 main categories of skills covered in the course (General Skills, Forensics & Steganography, Web Hacking/Exploitation, and Exploratory Bonus Topics). Student teams are expected to complete 1000 points worth of challenges per student in the team across a two-week period.*

Like CTF competitions, learners are prompted to develop and showcase engineering process skills. Teams draft and later present detailed technical writeups for each challenge, a practice that builds on their experience with the engineering notebook. Additionally, students are provided opportunities for metacognitive reflection through periodic collaborative work reflections. Here, they assess their own approach to teamwork and problem solving to identify strategies that work and ways to improve moving forward with their partner. This reflective practice equips learners with the tools to self-regulate their learning, fostering both individual growth and team cohesion.

This culminating project serves as both a practical introduction to the cybersecurity industry as well as a demonstration of skills developed throughout the course. By participating in a structured mock competition, students experience an authentic CTF competition setting which are commonly used for recruitment and skill assessment. The requirement for detailed technical writeups and their presentation during the final reinforces industry-standard documentation and communication skills. By combining technical proficiency with a supportive and engaging environment, the final project allows students to flex their skills in a relatively low-stakes yet meaningful context while preparing them for future academic and professional opportunities.

Assessment:

We designed *Introduction to Hacking Competitions* to provide numerous opportunities for formative assessment of student progress, allowing us to adapt content to where the individual student is at, even while they are working in a team to complete CTF challenges. All assignments demonstrate the learners' current understanding of a topic on their way to forming the *hacker mindset*. For example, prelab responses identify areas of confusion, allowing us to quickly adjust in-class content for the week. Meanwhile in-class activities are spaces for technical skill-building with more directed facilitation, where learners gain help from both the student-teachers and their peers. Perhaps most importantly, the engineering notebook makes learners' thinking visible, especially critical during CTF challenges when learners are working independently. In this way, we elucidate potential misconceptions while showcasing the current level of technical skill development, allowing both students and instructors to regularly reflect and check in on the learning process. Having this continuous feedback loop enables us to adjust our teaching strategies in real time, tailoring instruction to address areas where students may be struggling or

provide deeper insights into topics where students show interest and may be eager to learn more. By responding dynamically to student needs, we as instructors can create an engaging learning environment that fosters active participation and supports critical thinking.

We designed the culminating experience to summatively showcase the students' overall mastery of the course material as a mock CTF competition. Students demonstrate their achievement of intermediate-level technical skills through the completion of these challenges, each requiring specific skills to succeed. While our platform tracks attempted flag submissions (both correct and incorrect), the writeup is used to assess the level of proficiency in specific learning outcomes. Furthermore, the real-world ambiguity of these challenges asks student-teams to use their newly-developed *hacker mindset*. They apply critical analyses, diverse perspectives, and adaptable thinking to formulate their unique path to solution - showcasing proficiency in the soft skills needed to close the skills gap in industry. Together, these assessment strategies not only evaluate student performance but also encourage students to take active ownership in their learning and cultivate professional habits that will benefit students not only through their academic careers, but more importantly, beyond the classroom.

Furthermore, the act of teaching in itself amplifies learning. Through allowing students the opportunity and space to think critically and independently from their instructors, we open the door for increased peer collaboration. By comparing strategies and teaching their peers, students further deepen their understanding of cybersecurity concepts, improving retention while cultivating self-directed learning habits. At the same time, collaborative learning encourages peer-to-peer accountability and builds a strong sense of community within the course [7, 8]. These dynamics not only enhance individual growth but also align with the current expectations of the industry, where teamwork, the ability to effectively communicate technical solutions, and a readiness to adapt are essential.

Technical Specifications & CTF Challenge Platform Setup

To support experiential learning in the classroom, we designed cost-effective infrastructure that allows deep customization of CTF challenges. Based on our experience participating in cybersecurity competitions, we chose the current most popular platform, CTFd [10], to serve CTF challenges for this class. This platform is open-sourced, lightweight, and allows for modularity and customization using plugins and source modifications. The selection of CTFd was influenced by its growing adoption in academic environments and robust community support, particularly demonstrated by Arizona State University's successful implementation in their cybersecurity courses through what is now publicly known as The pwn.college Dojo [11].

From the beginning, we prioritized two key design principles: enabling continuous learning and minimizing clicks-to-content. The continuous learning aspect lets students access challenges at any time, allowing them to progress through the material at their own pace without depending on instructor intervention. Our "self-healing" infrastructure uses automated health checks and recovery procedures, removing the need for manual system administration for challenge resets or environment restoration. Automated provisioning systems detect and fix issues, keeping challenge environments running 24/7. The focus on reducing clicks-to-content removes common technical barriers like local sandbox setup, server configuration, and complex authentication that typically slow down learning. Unlike many cybersecurity labs and engineering courses with

complex lab setup procedures and resource scheduling, with our configuration, students can jump directly into hands-on work without dealing with infrastructure setup (beyond installation of the Kali Linux Distribution), allowing students to stay engaged with the content.

The current course curriculum, and thus the CTF challenges developed on this platform, centers on web exploitation, forensics, and Linux command line skills, implemented through a system of dynamic challenge instances. Each instance includes a completion token, or *flag*, that validates a student's successful application of cybersecurity skills - or records if students submit the incorrect flag. We built these instances using Docker containers, which create sandboxed environments to prevent destructive commands, enable quick restarts, and protect the host infrastructure from potential sandbox escapes. We picked Docker over Podman to manage all challenge instances under a single engine, prioritizing centralized control over individual container performance. This setup gives us precise control over challenge environments, letting us provide students with targeted information through environment variables and parameters (such as encryption keys students can uncover for their web exploitation challenges) as they work through each challenge. Docker's containerization has worked especially well for keeping challenge environments consistent while supporting the needed skill-building infrastructure.

This architecture extends beyond basic containment to focus on scalability. While the current course enrollment is limited to 20 students, our infrastructure framework is designed to handle hundreds of concurrent users. The system operates with a two-part design: a management system handling the web server, platform database, and container orchestration, paired with a second, dedicated system for hosting the containers themselves. We developed a lightweight Python plugin for the CTFd platform that connects to a Docker swarm, giving students the ability to start, stop, and restart their challenge instances through our orchestration engine. The setup leverages Docker's native swarm key management for authentication, and since both machines operate on the same local area network, it requires no special networking configuration. The platform's built-in backup and restore capabilities maintain system state and student progress during maintenance or updates, while the separation between management and execution systems prevents resource-intensive challenges from impacting core platform performance. **Figure 5** depicts this two-part platform architecture.

Future development focuses on expanding dynamic content generation beyond web and binary exploitation challenges. While our web exploitation instances give each student their own environment, file artifacts, such as network captures, are currently static across all users. We plan to build containers that generate unique files for each student, matching our current setup for web challenges. Storage space becomes a problem with this configuration since keeping unique files for every student takes up a lot of space, especially with larger class sizes. To fix this, we are developing a seed-based system where we only save the starting values used to make each student's files. The generation containers would use these seeds to remake the exact same content whenever a student downloads or resets their artifacts. Since our current artifact generation scripts can take several minutes to complete, we're implementing a caching layer that pre-generates and stores common artifacts. This cache would track student seeds and artifact versions, automatically regenerating cached content when scripts are updated while still serving cached versions for immediate download.

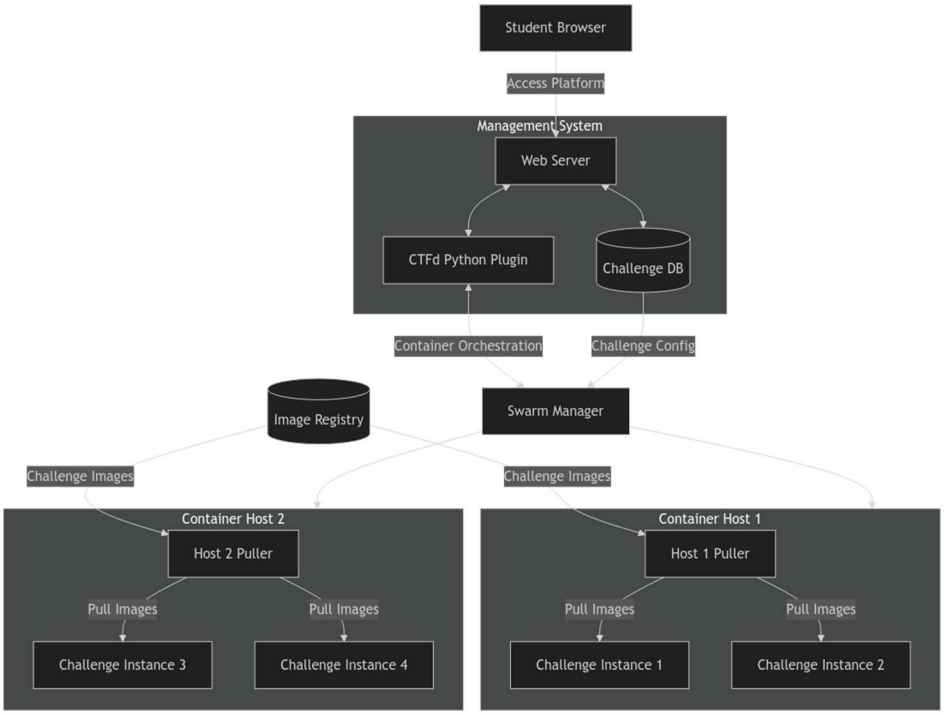

Figure 5*: The two main components of our infrastructure working together: the management system and container system. The management system serves as the central hub, handling both content delivery through the platform web server and orchestrating challenge instances for students. When students interact with the platform to request a challenge, our custom Python plugin seamlessly communicates with the platform's database system to gather essential challenge specifications, including the required Docker image and specific networking requirements. Following this request, our plugin establishes a secure tunnel connection with the Docker swarm, efficiently orchestrating container hosting across multiple systems. Once the necessary resources become available, the host system retrieves the specified image and initializes the container environment. Connection details are relayed back through our plugin to the student, providing secure, direct access to their personalized challenge instance.*

To support university maintainability of the system, we are looking into adding full virtual machine (VM) support alongside our container setup (depicted in **Figure 6**). While Docker works great for web and basic system challenges, some advanced topics need complete system access that containers can't provide. By adding Proxmox hypervisor support to our orchestration, we can give students full system images when needed. The integration would leverage Proxmox's API for automated VM provisioning and lifecycle management, using the native networking capabilities of the Proxmox Virtual Environment (PVE) to isolate student environments. Each VM would be assigned to isolated network bridges with custom firewall rules applied at the PVE level, preventing any traffic leakage between instances while still allowing internal networking for multi-machine challenges. The orchestration system would handle both container and VM lifecycles through a unified API layer, abstracting the underlying infrastructure differences from the CTFd plugin. This setup would include PVE firewall rules to restrict outbound connections and prevent students from accessing anything outside their assigned challenge network, while still allowing internal services like DNS and package mirrors to function. Initial testing shows this could work with minimal latency overhead, though storage requirements would need careful consideration as full VM images take up significantly more space than containers.

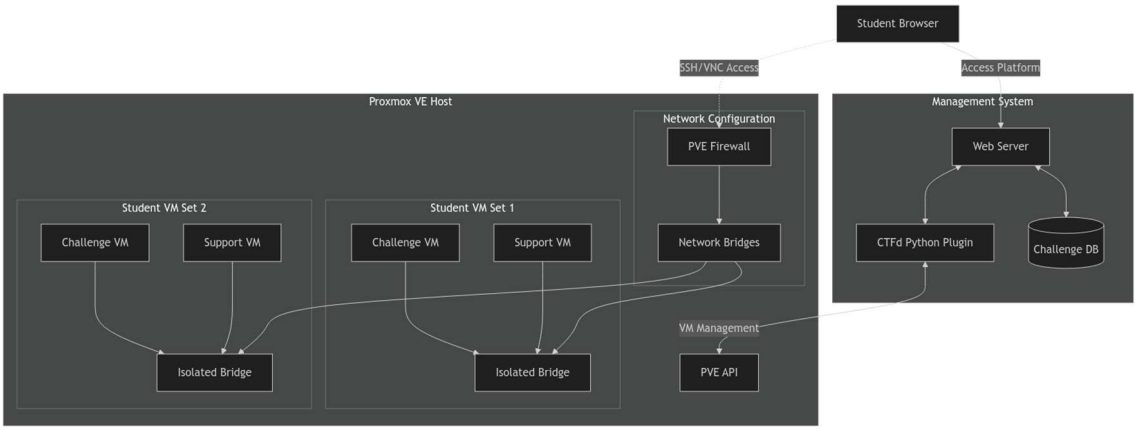

Figure 6*: This figure shows the potential addition of VM orchestration into our existing platform additions. Similar to the previously mentioned Docker connection, we can orchestrate the creation and management of VMs through a PVE API connection from our Python plugin running on the management system. With an existing network configuration on PVE, we can segment network bridges to secure groups of VMs to ensure students are only able to target their specific instances.*

Results and Conclusions

"*This is one of the best courses I have taken so far, its very engaging and I learned so much. The environment made it a room to learn and also there was so much collaboration that it was fun to be in the class. I personally have not taken any coding/computer related classes and was afraid that I wasn't going to know what to do but instead this is a very beginner friendly class and I am understanding everything very well.*" -Learner response in SETs

Introduction to Hacking Competitions is very successful overall in its goals of introducing interested students into the wide world of cybersecurity while also supporting learners new to the field in achieving intermediate-level, industry-applicable skills in cybersecurity, all in just over ten weeks. These outcomes are not only directly observed by the teaching team but are also reflected in the university administered Students' Evaluation of Teaching (SET), as well as externally-administered exit surveys (n=33) and lightly-structured interviews of the learners and student-teachers. All survey responses are kept anonymous from the teaching team. At the time of writing, this course has been fully offered twice, each showing significant interest in enrollment. Now in its third quarter, Spring '25 quarter's waitlist outnumbered the available slots three to one. Those enrolled in the course remain engaged throughout its 10 weeks; students go from nervously approaching the challenges to itching for more at the end. A major theme from learner feedback is the desire for more CTF challenges across all topics - especially the exploratory bonus topics. Students not only ask for more challenges but actually complete them, evinced in challenge completion rates: on average, student-teams achieve 25% more points during the final challenge than required for an A grade. At the end of the quarter(s), 60% of learners expressed strong interest in joining the university's SlugSecurity student club and continue working within this community of practice to advance their skills and experience.
"Thank you for teaching this class. It was a great experience, and I hope to attend Slug Security events when they align with my schedule. You guys are truly knowledgeable and are great teachers, especially for underclassmen." -Learner response from internal survey

A combination of descriptive statistics of quantitative survey items with inductive content analysis of the qualitative items highlight the key course features most valued by the students while elucidating their experience in the classroom. Emergent themes across the two offerings frame the class as "*great*;" learners also commonly use "*fun*" and "*enjoyable*" to describe their

learning experience. The data also shows a substantive increase in learner confidence in their technical hacking skills as well as their development of the hacker mindset, in alignment with industry needs (see Figure 7 in Appendix). Predominantly, students use open-ended survey items to express their appreciation for the class and its student-instructors, for example:
"Thanks! Y'all really made the class a lot of fun, and helped make it a good introduction to starting in Slug Security too. I really enjoyed this class." -Learner response from internal survey

When prompted to think about course structure and learning artifacts, the majority of students rated *all instructional features* Helpful or Very Helpful on 4-point Likert items, with a few notable exceptions (data given in Figure 8 in the appendix). As expected, learners highlight the CTF challenges as the most helpful aspect of this class for their learning, especially the *in-class challenges* as they "*let us see the thinking behind solving a problem*." Prelabs were rated lowest. Looking at the qualitative responses for more explanation, learners suggest prelabs can be better focused on more targeted and relevant content – though some explicitly called out prelabs as very helpful. As a whole, learners appreciate the course's high-structure as "*helpful*" and "*well-thought-out*." However, they also mention that the amount of material delivered in class can be toned down for better absorption, using words such as "*frustrating*" or "*confusing*." Moving forward, we aim to reevaluate and restructure prelabs and in-class time to more directly support targeted learning outcomes while balancing the spread of concepts over time.
"*The instructors gave great examples and were really good at making the topics interesting.*"
- Learner response from SETs.

Our analysis of class climate depicts a welcoming and accessible learning environment for early-career students. A minimum of 87% of learners responded Yes or Definitely Yes (4 or 5 on 5-point Likert) to *every question* gauging their comfort in the classroom, a concept broken into the following question categories: diversity, inclusivity, and interactions with student-instructors.
"Awesome class! My favorite I've taken at this school so far. All really new information and the activities were really fun. I always look forward to coming to class. /srs" - Learner response from internal survey.

Preliminary results from questions targeting affect towards pursuing an engineering major show that this class significantly affected learners' academic goals, either by steering learners that had not yet declared a major into engineering (22% of respondents) or by reaffirming learners' decision to declare an engineering major (46% of respondents).
"*Yes, it just amplified my interest in cybersecurity from "its cool" to "why isn't there a major/minor/ concentration"* -Learner response in internal survey.

Overall, students across course offerings all reported positive experiences and enjoyed the peer-led collaborative structure facilitated by the Early Design Experiences program. Although there is room for future improvements, the current iteration of *Intro to Hacking Competitions* succeeds in its role of realizing a supportive and engaging learning environment, providing beginners a safe space to develop and hone their skills before graduating into the wider cyberspace. Looking ahead, the continued evolution of this course, driven by student feedback and demands of the cybersecurity industry, has the potential to further define its impact and better prepare future cohorts for the challenges and opportunities they will encounter in the cybersecurity field.
"*I don't know what to say besides Ian and Jacobs are goated at teaching. The lectures were great and the overall design of the course felt very natural and the amount of information covered was perfect. I feel like I not only learned a lot of things that I can apply now, but also learned ways to continue to grow my knowledge in cybersecurity*." - Learner response from SETs.

Closing sentiments from the student-teachers:

"As a student-instructor who has approached this field from the outside and is a relative newcomer, the drive of the students and their improvement observed over the course of the 10-week academic quarter was genuinely very impressive. As someone who has been in these students' shoes not too long ago, seeing the rapid growth and development of skills was quite remarkable, and as someone who started in the field as a beginner and struggled many of times to get to the level at which I am today, I wish that I had something like this course to guide me along the way."- **Ian Phan**

"From someone who works within the cybersecurity sector, we observed students being very dependent on guidance from instructors at the start of the quarter, unsure how to go about researching for help. At the conclusion of the course, students had developed skills that matched what you'd expect after a cybersecurity internship - they were self-reliant, could troubleshoot effectively on their own, and knew how to find the right resources to solve technical problems. This growth in independence was extremely impressive given the 10-week course where most students had minimal experience with the content." - **Iakov Taranenko.**

Appendix

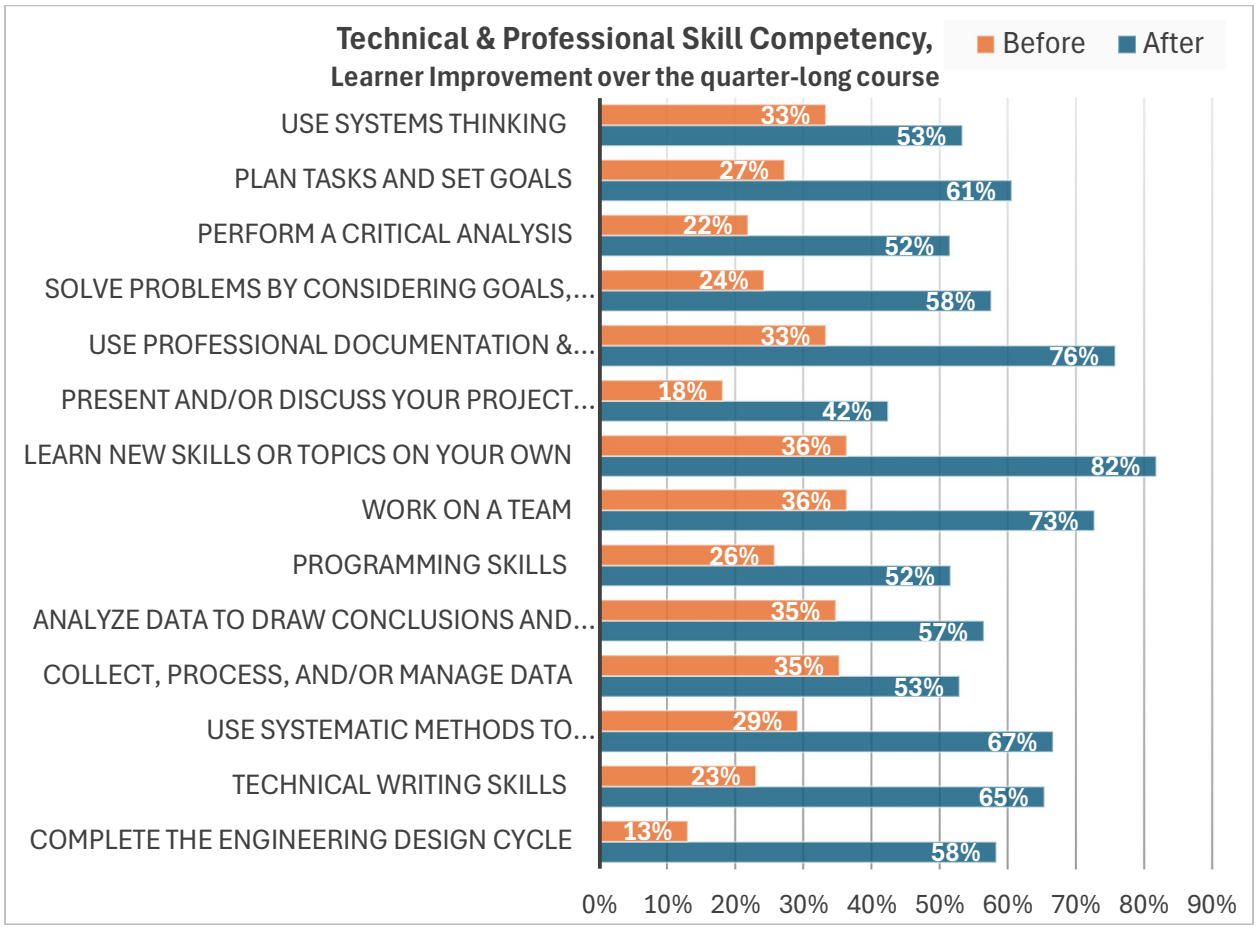

***Figure** 7: Student responses to 5-point Likert survey items asking them to rate their own skill-level in areas related to the primary learning outcomes of Intro to Hacking Competitions. In this figure, the data shows the percentage of students rating themselves as Good or Very Good as a single bin. Skills shown in orange are their perceived rating at the beginning of the quarter while the values shown in blue reflect student achievement and confidence at the end of the quarter. Note only selected skills that reflect the students' development of the hacker mindset are shown here from the survey data. All learning goals display substantive improvement among students.*

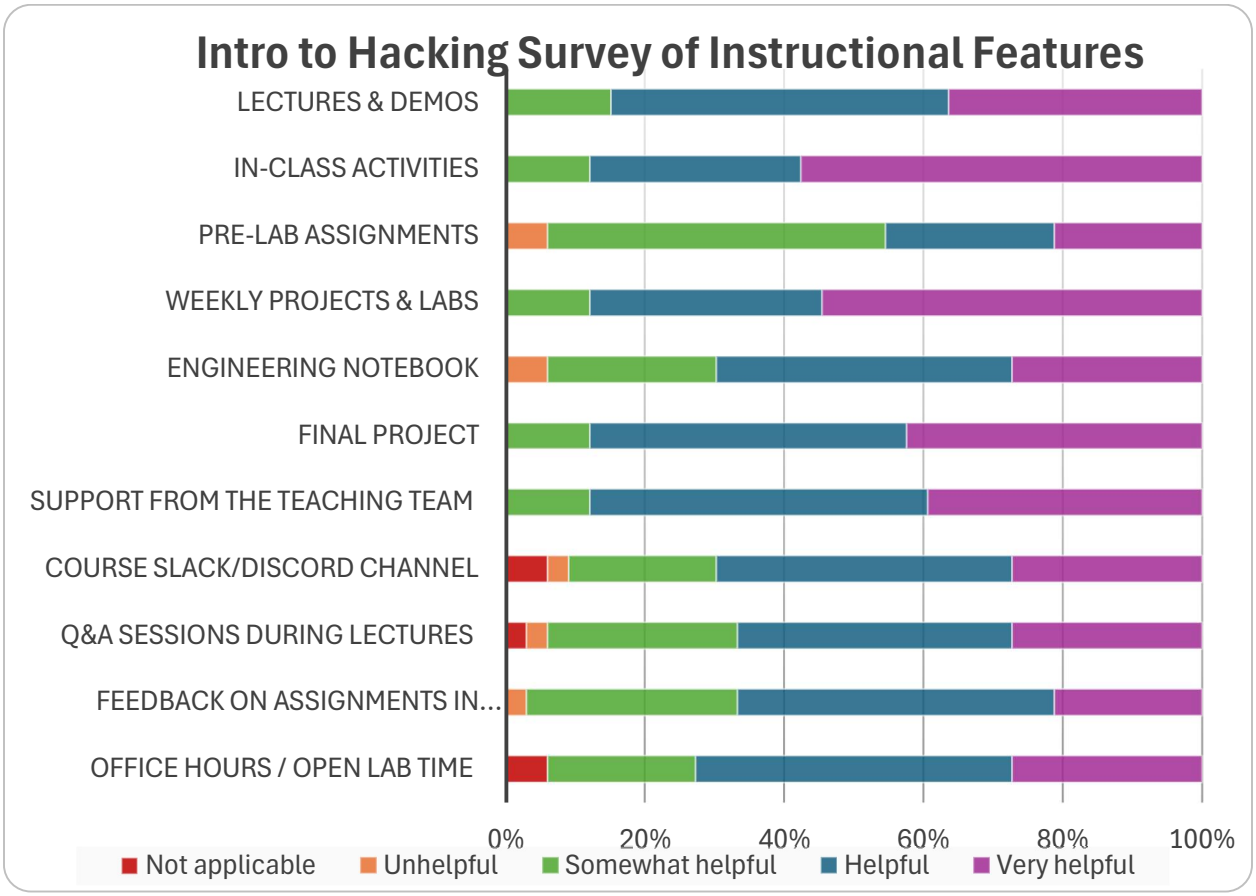

Figure 8: *Student responses to 4-point Likert survey prompts evaluating the helpfulness of the course learning artifacts. Orange = "Unhelpful," Green = "Somewhat Helpful," Blue = "Helpful," and Purple = "Very Helpful." Note that artifacts that learners perceived as not applicable to their course are displayed in red. Students highly rated the active and hands on features of the course, namely the CTF challenges.*

References

1. "Employers must act as cybersecurity workforce growth stalls and skills gaps widen," Cybersecurity Certifications and Continuing Education, https://www.isc2.org/Insights/2024/09/Employers-Must-Act-Cybersecurity-Workforce Growth-Stalls-as-Skills-Gaps-Widen.
2. N. Eddy, "Cybersecurity talent shortage prompts White House action," Cybersecurity Talent Shortage Prompts White House Action, https://www.darkreading.com/cybersecurity-operations/cybersecurity-talent-shortage prompts-white-house-action.
3. "Cybersecurity supply and demand heat map," Cybersecurity Supply and Demand Heat Map, https://www.cyberseek.org/heatmap.html.
4. "Cybersecurity career opportunities outpace supply, new CyberSeek Data reveals," CompTIA, https://www.comptia.org/newsroom/press-releases/cybersecurity-career opportunities-outpace-supply-new-cyberseek-data-reveals.

5. Global Cybersecurity Forum and Boston Consulting Group, “The 2024 Cybersecurity Workforce Report,” 2024. [Online]. Available: https://web-assets.bcg.com/61/d3/705fbd684d70b0e5f98cdcf7cf47/2024-cybersecurity-workforce-report.pdf.
6. K. Owens, A. Fulton, L. Jones, and M. Carlisle, *pico-Boo!: How to avoid scaring students away in a CTF competition*, 2019.
7. E. Wachtel, Q. Cao, M. Kaltman, K. Tran, M. Robles Hernandez, and T. Favaloro, “Board 174: Fostering Inclusivity and Engagement while Learning by Doing: A New Paradigm in Engineering Education Based on Student-Designed, Student-Taught Courses,” in *2024 ASEE Annual Conference & Exposition Proceedings*, Portland, Oregon: ASEE Conferences, Jun. 2024, p. 46737. doi: 10.18260/1-2--46737.
8. T. Favaloro, “Building the Engineering Identity of the Lower-Division Engineer: A Formal Model for Informal Peer-to-Peer Mentorship and Student Leadership through Undergraduate Student-Led Experiential Learning,” in *2024 ASEE Annual Conference & Exposition Proceedings*, Portland, Oregon: ASEE Conferences, Jun. 2024, p. 48430. doi: 10.18260/1-2--48430.
9. T. Favaloro, "Students as the Teachers: Positioning Undergraduates as Experts, Role-Models, and Guides to Create Diverse Learning Communities," *2024 IEEE Frontiers in Education Conference (FIE)*, Washington, DC, USA, 2024, pp. 1-9, doi: 10.1109/FIE61694.2024.10893309.
10. CTFd: CTFs as you need them, http://github.com/CTFd/CTFd (accessed Jan. 15, 2025).
11. Pwncollege/dojo: Infrastructure powering the pwn.college dojo, https://github.com/pwncollege/dojo (accessed Jan. 15, 2025).