Remote Controlled Cyber: Toward Engaging and Educating a Diverse Cybersecurity Workforce

Curtice Gough
cgough2019@my.fit.edu
L3Harris Institute for Assured Information
Florida Institute of Technology
Melbourne, FL, USA

Carl Mann
cmann2013@my.fit.edu
L3Harris Institute for Assured Information
Florida Institute of Technology
Melbourne, FL, USA

Cherrise Ficke
cficke2018@my.fit.edu
Florida Institute of Technology
Melbourne, FL, USA

Maureen Namukasa
mnamukasa2020@my.fit.edu
Florida Institute of Technology
Melbourne, FL, USA

Meredith Carroll
mcarroll@fit.edu
Florida Institute of Technology
Melbourne, FL, USA

TJ OConnor
toconnor@fit.edu
L3Harris Institute for Assured Information
Florida Institute of Technology
Melbourne, FL, USA

ABSTRACT

Cybersecurity education has grown exponentially to support the need for a skilled cybersecurity workforce. Further, capture-the-flag competitions have popularized cybersecurity by engaging and recruiting students while exposing them to cybersecurity workforce competencies. However, the heavy reliance on competition-based educational approaches may contribute to the lack of diversity in cybersecurity programs. Cybersecurity competitions are the primary catalyst to expose and recruit students from both high school and collegiate cybersecurity education programs. In response, we propose a collaborative, experiential learning approach that leverages hackable Internet of Things (IoT) toys as a pedagogical tool for cybersecurity education. We share our detailed design, activities, experiences, and lessons learned for others to build on our initial success.

CCS CONCEPTS

• **Social and professional topics** → **Model curricula**; **Computing education programs**.

KEYWORDS

cybersecurity education, workforce competencies, diversity

ACM Reference Format:
Curtice Gough, Carl Mann, Cherrise Ficke, Maureen Namukasa, Meredith Carroll, and TJ OConnor. 2024. Remote Controlled Cyber: Toward Engaging and Educating a Diverse Cybersecurity Workforce. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024), March 20–23, 2024, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630917

SIGCSE 2024, March 20–23, 2024, Portland, OR, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0423-9/24/03.
https://doi.org/10.1145/3626252.3630917

1 INTRODUCTION

Capture the Flag (CTF) games are pervasive in education settings as teaching and engagement strategies [12, 13, 16, 20, 26, 27, 31, 33, 37, 44, 46, 48, 50]. Previous works have argued that CTFs increase student learning outcomes, develop workforce competencies, and deliver enjoyable assessment methods [20, 48]. Further, CTF-based approaches provide an opportunity to increase participation in the growing cybersecurity discipline. CTFs have proven effective at recruiting individuals interested in cybersecurity; however, they fail to attract and engage historically marginalized groups without previous exposure [24]. Statistics from national and international CTF competitions hint that CTFs fail to engage and recruit a diverse cohort of cybersecurity students and professionals. Demographics from the 2023 High School Cyber Patriot program reveal that only 27% of competitors identify as female and 7% as African American [2]. Further, these demographics have remained stagnant for five years, with female and African American participation growing only 2% and 1%, respectively.

We hypothesize that the current cybersecurity educational approaches that heavily rely on the capture-the-flag (CTF) style gamification model may unintentionally reinforce the barriers for underrepresented groups in the cybersecurity domain. CTF games often map to workforce competencies [36], and gamification is a highly effective learning tool across several disciplines. However, previous work suggests that competition-focused gamification may not equally benefit all individual learners [17]. There may be negative impacts on underrepresented minorities in competitive environments [39]. In contrast, supportive and collaborative learning environments may be more beneficial.

In the following work, we propose a collaborative, experiential learning approach that leverages hackable toys as a pedagogical tool for cybersecurity education. Further, we embrace scaffolding and direct mentoring as supportive strategies to engage historically marginalized groups in cybersecurity. Our approach presents an opportunity to overcome the artificial roadblocks created by the growing emphasis on CTF competitions. This paper makes the following contributions:

(1) We share our detailed design, activities, experiences, and lessons learned for a collaborative, experiential learning approach that leverages hackable toys as a pedagogical tool for cybersecurity education.
(2) To allow other instructors to build on our initial success, we publish our classroom slides, lecture materials, activities, and code to reproduce at https://github.com/tj-oconnor/Remote-Controlled-Cyber.

Organization: Section 2 investigates the workforce competencies and gamification strategies for cybersecurity. Section 3 provides an overview of our approach, platform, and modular design. Section 4 examines the design of our modules. Section 5 explores our spiral development model and our test groups. Section 6 offers insight and examines future challenges. Section 7 explores opportunities for future work. Section 8 summarizes our findings.

2 BACKGROUND

The following section motivates the problem by examining cybersecurity workforce competencies and pedagogical approaches for diversity and gamification.

2.1 Cybersecurity Workforce Frameworks

A growing demand exists for professionals with cybersecurity education. However, the discipline of cybersecurity is often vaguely and broadly described. Existing frameworks, accreditation, and curricular guidance often present competing guidance prioritizing different cybersecurity domains. We examined this guidance before determining the topics for our activities. The National Initiative for Cybersecurity Education (NICE) publishes a framework of fundamental cybersecurity knowledge, skills, and abilities [36]. This framework focuses broadly on several non-technical areas, including planning, policy, governance, and management. This framework further creates a technical taxonomy of knowledge, skills, and abilities. In contrast, the National Security Agency (NSA) provides a curriculum accreditation requirement that focuses on deep technical concepts in computer science, computer engineering, and electrical engineering and places a higher emphasis on binary reverse engineering and vulnerability research topics [25]. This approach benefits the development of offensive cyber operations tools necessary to operate in cyberspace. Finally, the Association for Computers and Machinery (ACM) circulates comprehensive and flexible curricula in cybersecurity education that broadly prescribes knowledge units across data, components, connection, software, and organizational security [1]. This curriculum emphasizes domains like system testing that support developing secure and reliable code.

2.2 Related Work

Gamifying Cybersecurity Education: Previous works have explored the benefits of gamification in cybersecurity education [12, 13, 16, 26, 31, 33, 37, 44, 46, 50]. These approaches have heavily relied on classroom capture-the-flag (CTF) competitions to encourage and engage students in cybersecurity [12]. In developing Aquinas, Petullo argued that cybersecurity students must have solutions capable of self-teaching and immediate feedback [37]. Burns et al. analyzed 3,600 Capture the Flag challenges and identified they focus on five key topics: cryptography, penetration testing of web vulnerabilities, reverse engineering, forensic analysis, and binary exploitation [11]. Švábenskỳ et al. identified that despite a focus on network security monitoring and ethical hacking, few classroom approaches introduce an adversarial mindset [44, 45]. Our previous work observed that cooperative team learning could balance the negative impacts of gamification [33]. We extend this work to identify opportunities to educate cybersecurity workforce-centric skills in an engaging methodology. Inspired by the clarity of CTF problems, we developed hands-on cooperative activities that span various NICE knowledge skills, abilities (KSAs), NSA knowledge units (KUs), ACM Curriculum, and CTF categories. Table 1 depicts our mapping of these activities. Section 4 expands on each exercise, providing an overview of the activity, the learner outcomes, and the engagement strategies.

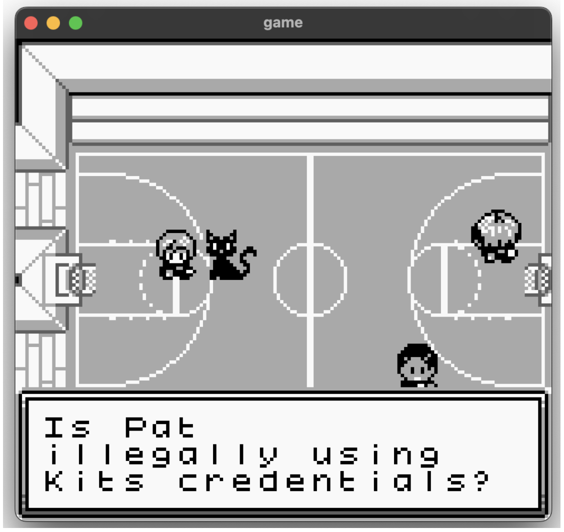
Diversity Approaches: Osman et al. conducted interviews with underrepresented minorities' interviews to understand the factors that attracted them to cybersecurity, how they built their skills, overcame hurdles, and maintained engagement [35]. Their work identified 19 recommendations for practice to engage a diverse cybersecurity workforce. We highlight three of these recommendations. First, we examine how to *Bolster Self-Efficacy* The lack of traditional education resources demands that cybersecurity students aspire to become autodidacts responsible for their learning [8, 10]. Progressing towards this goal, we incorporate scaffolded code and achievable modules to foster self-belief [21]. Next, we explore how to *highlight the societal relevance of cybersecurity.* We design each module with a lecture exploring cybersecurity's historical impacts. For example, we motivate our binary exploitation module by examining how STUXNET sabotaged Iran's uranium enrichment program to prevent nuclear weapons development [19]. This approach allows students to see meaningful, real-world impacts. Finally, we design each module to *integrate active problem-solving exercises.* Each module consists of a challenge that teammates must solve collaboratively. Collaboration has been shown to benefit female students that tend to be more mastery-oriented [5, 41]. For example, during the *Grand Theft Crypto* module, one student must brute-force values while another observes the outcomes. We believe the emphasis on collaboration is necessary to engage historically marginalized groups.

3 OVERVIEW

As depicted in Table 1, our modules expose students to five ACM Curriculum areas, twenty NICE workforce competencies, and eight NSA CAE-CO knowledge units. We developed the modules to target a high-school based audience. We leverage this broad-based approach to engage and educate a diverse workforce by relying on the best practices [35]. Each module includes a lecture that introduces technical material and highlights the societal relevance of that domain. Following each lecture, we deliver a collaborative activity integrating active problem-solving exercises. Further, we designed the activities with appropriate digital prompts and scaffolded solutions. This careful approach ensures challenge-skill match, leading to bolstering efficacy in learners. In the following section, we explore the modules in depth.

Lesson	Platform	CTF	ACM	NICE	NSA CAE-CO
Ethics or Death	Game Boy	-	Cyber Ethics	K0003, K0524, A0046	Legal and Ethics
King of The Packet	RC Car	Forensics	Network Services	K0001,K0362,S0065	Networking
Attack Oriented Prog.	Game Boy	-	System Testing	S0019,S0130,S0266	Systems Programming
Beating Rumpelstiltskin	Game Boy	Reverse Engineering	-	K0175,K0183	Software Reverse Engineering
Grand Theft Crypto	RC Car	Cryptography	Cryptography	K0019,K0190,K0403	Applied Cryptography
Hack This Car	RC Car	Web Vulnerabilities	Software Security	K0398,K0624,A0092	Software Security Analysis
Pwn My Ride	RC Car	Pwn	-	K0070,S0014,S0293	Vulnerabilities

Table 1: Our modules deliver cooperative activities that span various ACM Curriculum, NICE knowledge skills, abilities (KSAs), NSA knowledge units (KUs), and Capture-The-Flag categories



Is Pat illegally using Kits credentials?

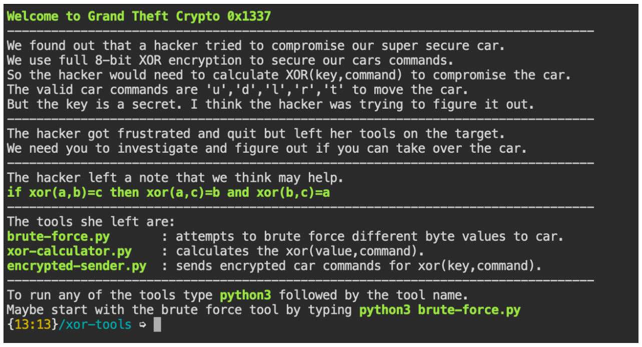
Figure 1: Our approach begins with a Game Boy game that introduces students to cyber ethical and legal dilemmas.

4 MODULES

In the following section, we explore how our modules bolster self-efficacy, highlight the societal relevance of cybersecurity, and integrate active problem-solving exercises toward engaging and educating a diverse cybersecurity workforce.

Ethics or Death: As discuss in [6], all cybersecurity education should address ethics first. Although studying offensive cybersecurity is engaging and exciting, critics are often against such approaches due to the possibility of misusing skills [14]. As such, we begin our modules with an ethical foundation for all other modules to build on. We focus our *Ethics or Death* module on developing students' moral sensitivity [40] to cybersecurity ethical and legal issues. We began the module with a lecture on key cybersecurity laws, including the Computer Fraud and Abuse Act (CFAA), Digital Millennium Copyright Act (DCMA), Electronic Communication Privacy Act (ECPA), and Access Device Statute. Following the lecture, we provide students with a replica Game Boy handheld console running a homebrew game. As depicted in Figure 1, the students move through a series of levels where characters introduce various ethical dilemmas. If the player fails to identify ethical issues, the game sends them to jail, where they must retrain. In our initial testing, we encountered an exciting ethical pivot worth discussing. Initially, we programmed the game so that correct answers were always the first choice of a multiple-choice question. After identifying this issue, we deliberately chose to keep it and use it as a teachable moment following the activity. We followed through after the activity, asked students if they identified the flaw, and discussed the ethical issues surrounding cheating. Ultimately, we developed this module to establish a solid foundation for the subsequent series of adversary-oriented modules by nurturing students' moral sensitivity through practical experience. In future course iterations, we will introduce and explore responsible disclosure by sharing narratives of our experiences in reporting IoT vulnerabilities to industry [18, 28–30].

King of the Packet: Our *King of the Packet* module explores network traffic analysis. We begin the lecture by introducing networking and exploring a historical example of how attackers compromised the telemetry of a drone. After motivating the problem, we move to a hands-on activity on a remote-controlled car. CTF competitions often include network traffic analysis challenges that require competitors to parse multimedia content embedded in network traffic [11, 12]. In this delivery model, network forensic challenges are often considered defense-oriented and do not benefit from engagement strategies that leverage adversarial thinking [15, 31, 45]. Similar to this approach, we captured the remote control car's network traffic and asked students to decode the traffic to parse the protocol, network address, port, login credentials, and embedded commands. However, we then challenged the students to embrace an adversarial thought process by asking them to take control of their car. In our classroom experiments, we often extended this activity to a king of the hill competition by placing students into groups surrounding a car under a traffic cone. We challenged the students to drive their cars to the cone and stay connected to the cone for two minutes. To further embrace the adversarial thought process, we prompted students by letting them know they could attack the other teams' cars or the car with the cone. As expected, this often turned into a chaotic event as teams pivoted between driving their car, the opposing group's car, and the car under the cone. As expected, few teams succeeded in staying connected for over two minutes due to the constant context switching. By creating an unwinnable activity, we delivered an engaging activity that avoided the negative impact of competition.

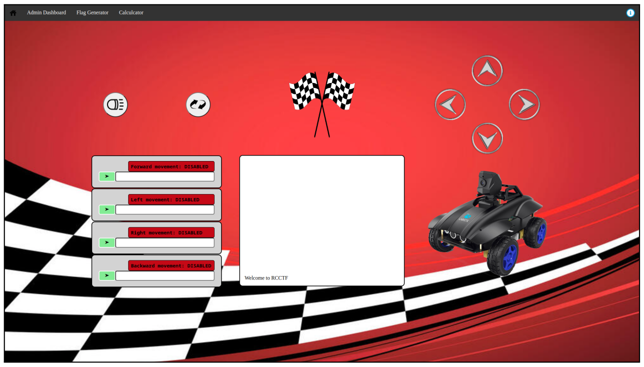


```
Welcome to Grand Theft Crypto 0x1337
------------------------------------------------------------------------
We found out that a hacker tried to compromise our super secure car.
We use full 8-bit XOR encryption to secure our cars commands.
So the hacker would need to calculate XOR(key,command) to compromise the car.
The valid car commands are 'u','d','l','r','t' to move the car.
But the key is a secret. I think the hacker was trying to figure it out.
------------------------------------------------------------------------
The hacker got frustrated and quit but left her tools on the target.
We need you to investigate and figure out if you can take over the car.
------------------------------------------------------------------------
The hacker left a note that we think may help.
if xor(a,b)=c then xor(a,c)=b and xor(b,c)=a
------------------------------------------------------------------------
The tools she left are:
brute-force.py        : attempts to brute force different byte values to car.
xor-calculator.py     : calculates the xor(value,command).
encrypted-sender.py   : sends encrypted car commands for xor(key,command).
------------------------------------------------------------------------
To run any of the tools type python3 followed by the tool name.
Maybe start with the brute force tool by typing python3 brute-force.py
{13:13}/xor-tools ⇨ █
```

Figure 2: Each module provides a prompt, accessible by a web browser, that delivers the scaffolding to guide learning.

Grand Theft Crypto: In our *Grand Theft Crypto* module, we extended the earlier *King of the Packet* activity by creating a symmetric encryption scheme for the car. The scheme encrypted the car's commands with a secret key known only to a legitimate driver and the engine. After lecturing students on data representation, cryptography, encryption, and cryptography attack approaches, we challenged the students to gain unauthorized control of the car. As depicted in Figure 2, we provided scaffolded code to allow the students to brute-force through a range of possible messages, encrypt a message given a key, and send encrypted traffic to the car. We placed obvious flaws into the design of the encryption scheme, limiting the necessary range to brute-force to 256 possible messages. Further, our encryption scheme relied on a one-byte key XOR'd with the plaintext message. As students brute-force and sent ciphertext messages to the car, they examined any vehicle movement to recover the original plaintext message. With an understanding of the plaintext and ciphertext messages, students could replicate a known-plaintext attack to recover the key. During this module's development and initial testing, we observed students solved this problem differently. Instead of relying on a known-plaintext attack to uncover the key, some students performed a replay attack by resending encrypted messages that moved the car. This observation demonstrates a vital issue in cybersecurity: several paths to success exist. Further, this approach allows advanced students to explore the breadth of solutions. We discuss this opportunity further in Section 6.

Attack-Oriented Programming: Our *Attack-Oriented Programming* module introduces key programming concepts, including variables, selection, iteration, and execution. Specifically, we present the high-level, interpreted Python3 programming language due to widespread adoption in cybersecurity tools. After the lecture introduces these concepts, we invite students to participate in an activity on the modified Game Boy. The modified Game Boy hosts an emulator that executes our homebrew Game. Further, the modified Game Boy hosts a Linux terminal accessible via a web server. The homebrew game challenges students to solve problems by programming in the terminal at specific points. To solve these challenges, students must write Python3 code that executes another program repeatedly in a loop until meeting a termination condition. Correctly solving the challenge allows the student to progress to the next level in the Game Boy game. As we examine in Section 6, this module introduced the most significant difficulty to students. As a result, we made several changes to our scaffolding that eased the challenge of systems testing.



Forward movement: DISABLED

Left movement: DISABLED

Right movement: DISABLED

Backward movement: DISABLED

Welcome to RCCTF

Figure 3: In our *Hack-This-Car* lesson, students work collaboratively to compromise web vulnerabilities, unlocking the control of a remote-controlled car.

Beating Rumpelstilkin with Z3: The *Beating Rumpelstilkin* module introduces the concept of constraint solving. Constraint-solving is a key component of software reverse engineering. Further, understanding constraint solving simplifies understanding advanced reverse engineering domains like symbolic execution. We begin the module with a lecture on constraint solving and motivate the problem with the historical example of the Conficker Working Group's reverse engineering the worms domain generation algorithm [38]. We introduce the Z3 open-source theorem prover during the lecture. In addition to other capabilities, Z3 provides a powerful ability to solve constraint problems and offers an easy-to-use Python3 API. For the activity, we present a Game Boy game where the student must pass levels by submitting the results of complex equations. Like the previous module, the student can pivot to a terminal providing tools and scaffolded code. We purposely placed this module after the *Attack Oriented Programming* module to justify the importance of learning programming in students' cybersecurity exploration.

Hack This Car: We designed the *Hack This Car* module around the concept of a Bug Bounty to release the power and steering controls of our remote controlled car. In recent years, companies have incentivized cybersecurity reporting by creating organizational bug bounty programs encouraging security researchers to disclose vulnerabilities in exchange for a monetary reward [9, 23]. Researchers identify bugs by systematically analyzing source code and configuration data [36]. Typically, bug bounty programs focus on web-based platforms that are publicly available for researchers to investigate. Centralized reporting platforms like HackerOne allow researchers to disclose vulnerabilities without fear of retaliation [23]. In response to the popularity of bug bounties, several prominent researchers have begun offering bug bounty courses as an alternative educational approach to teach students and researchers about the methodology and tooling necessary to investigate bugs [3]. Before the activity, we provide students with a lecture on web vulnerabilities. As depicted in Figure 3, the activity includes prompts to direct the student to vulnerabilities in the web application. For example, the application contains a boolean value cookie that sets the state of the administrator. By manipulating this value to True, the students can move the car in the left direction. Students gain complete control of the car by solving all the bounty challenges.

Pwn My Ride: Our *Pwn My Ride* module examines the concept of buffer overflows, dynamic analysis, and binary exploitation. In this activity, we explained to students that the car application, used in previous modules, had been disabled by deliberating removing the steering control function call. However, in their haste, the developers also used a vulnerable input call that failed to validate input, allowing for a buffer overflow. Similar to the previous lesson, we challenge students to uncover and exploit a security flaw. After a lecture on buffer overflows and debugging, we challenged the students to restore their access to the steering controls by overflowing the input, gaining control of the program counter (PC) register, and pointing it at correct function address. By allowing the students to compromise the car, we introduce the severity of binary exploits. As reverse engineering and binary exploitation are rarely taught and embraced in classroom environments [4, 31, 32, 34]; we approached this module carefully, creating both novice and advanced modules. The advanced modules extend the study to include understanding how the prologue and epilogue of Aarch64 functions and memory register purposes affect code-reuse attacks like return-oriented programming.

5 EVALUATION

We leveraged a spiral development approach [7] in which we developed, delivered, and evaluated modules to various demographics. This approach allowed us to refine several key variables and test different educational strategies. For example, in Section 6, we describe how this strategy informed us about the appropriate amount of scaffolding to achieve guided discovery throughout the various modules. Over the previous year, we conducted trial lectures and activities with the following groups.

- Middle School Honors Math Class
- High School Army Junior ROTC Class
- High School Air Force Junior ROTC Class
- Mixed Audience Expertise, B-Sides Security Conference
- Mixed Audience Expertise, Novice Cybersecurity Club
- Undergraduate Students, Cybersecurity Degree
- Mixed Audience Expertise, Aeronautics Graduate Students

For a portion of these evaluations, we obtained IRB approval and collected empirical data regarding student engagement and challenge levels experienced during the lessons, as well and changes in student intent to pursue a cybersecurity career or education occurring from pre to post exposure to the lesson. Results indicated that: (a) participants experienced a moderate amount of challenge, so the modules were at the right difficulty level; (b) all participants experienced high levels of engagement, with underrepresented minorities (URMs) reporting significantly higher engagement than non-URMs; and (c) significantly more female than male students reporting increased levels of intent to pursue cybersecurity after participating in the lesson (Namukasa et al., under revision). We also collected qualitative data regarding student positive and negative reactions to the course that served as lessons learned.

6 LESSONS LEARNED

The following section shares our challenges and successes in developing our modules and pedagogical approach that leverages hackable toys to engage and educate.

6.1 Challenges

Embracing the Digital Divide: We encountered an interesting observation when presenting our *Grand Theft Crypto* module to a Middle School Honors Math Class. Although students thoroughly participated and accomplished the outcomes of the module, the teacher struggled when the module pivoted to the hands-on activity. Self-admittedly frustrated with their lack of understanding, the teacher asked the students how they accomplished the activity outcomes. After examining the situation, we realized that the students were part of Generation Alpha, who had grown up with ubiquitous computing and networking [47, 51]. These digital natives felt far more comfortable on mobile devices than their digital immigrant Generation X teacher. Without their teacher's guidance or prompting, the students responded to the digital prompts on their mobile devices to complete the activity. While adept at technology, the digital immigrant teacher felt far less comfortable approaching newer technology. Following this insight, we iterated our design process to ensure all activities had digital prompts to allow the students to engage in self-learning.

This Scaffolding Is Just Right: Previous work has investigated the amount of scaffolding necessary for discovery by computer science students [21, 49]. Correctly balancing scaffolding results in guided discovery with students demonstrating proximal flow learning [49]. Previous work has shown this approach beneficial, even for teaching complex reverse engineering concepts like symbolic execution [42]. In our evaluation, we explored different amounts and types of scaffolding, observing student responses. For example, during the activity at B-Sides, we provided challenges with relatively few digital prompts or starter code. This resulted in a limited pool of novices trying the challenges. We saw this pattern repeat with the *attack oriented programming* module, where students struggled with syntactical problems like Python3 whitespace indentation. We iterated this design to develop digital prompts or starter code examples that bypassed these syntactical challenges. We then tested this improved design with a trial with novice students at our university, observing they enjoyed attacking the problem instead of the syntax. Our spiral design approach demands further formal study to identify the optimal amount of scaffolding for each module and reserve a formal curriculum evaluation for future work.

6.2 Successes

Hiding the Easter Eggs: In video games, Easter eggs provide undocumented features often hidden by the game's developer. One of the earliest and most well-known Easter eggs is the Konami Code, a sequence of inputs that yield hidden features [22]. Initial work has shown hidden content to prove an effective strategy in

```
    if msg == b'f' or msg == b'w':
        car.forward()
        sock.sendto(b'Going␣forwards\n', client)
    elif msg == b'b' or msg == b's':
        sock.sendto(b"Going␣backwards\n", client)
        car.backward()
    ...
    elif msg == b’spin’:
        sock.sendto(b"Spin␣cycle...\n", client)
        car.spin()
    elif msg == b’dance’:
        sock.sendto(b"Jiggle␣jiggle\n", client)
        car.dance()
```

Figure 4: We hid easter eggs in each module to create effective dialogue that empowered learners.

the classroom by communicating the teacher is seeking innovative communication mechanisms [43]. As such, we explored this concept by adding several Easter eggs to our hackable toys. As depicted in Figure 4 our car has undocumented features, including lights, spin, or dance, that the students can identify by reviewing the source code. In our trials, we discovered that leaving these hidden features proved a helpful pivot when students exceeded the course outcomes faster than their peers. As explored earlier, this allowed us to create an effective dialogue and empowered learners. Our early evaluations showed that these Easter eggs developed positive relationships between students and teachers. This approach established a hidden secret, empowering students to explore the module deeper instead of reaching an arbitrary terminal conclusion of the activity.

Taking The Unintended Path: Cybersecurity educational approaches often encourage creativity by emphasizing the solution instead of following a process [10]. The ever-evolving landscape of cybersecurity demands individuals who can think creatively and purposely deviate from the directed paths or processes [8, 15, 26]. To reinforce this paradigm, our modules contain multiple paths to success. While this concept is intuitive to hackers, it proves extremely difficult to understand for our colleagues in other domains who demand convergent solutions. Our colleagues in other disciplines would often ask, *What is the correct solution?* and become frustrated with our response *Any solution that achieves the outcome.* We observed one of these deviations while testing the *Grand Theft Crypto* module. When designing the module, we anticipated students would approach the solution by calculating the key from XOR of an observed action and brute-forced value. However, a middle-school student identified a second path during our testing. Her solution achieves the same outcome without discovering the key by replaying the brute-forced value. Both approaches successfully reach the outcome of moving the car forward. As the learning outcome of the module is to introduce and apply a cryptographic attack, both paths reach the same learning outcome. This approach further allows us to direct path exploration and pivots for advanced students who develop solutions before their peers.

7 FUTURE WORK

Our work introduces the idea of using IoT toys as a pedagogical tool for cybersecurity education. As such, we share our initial findings developed during our spiral development approach. We reserve future work to explore a more detailed experiment, examining how this approach affects demographics of gender and race.We reserve this future work to explore how our approach affects learner outcomes.

8 CONCLUSION

In this paper, we presented our collaborative, experiential learning approach that leverages hackable Internet of Things (IoT) toys as a pedagogical tool for cybersecurity education. Our modules build on the best practices for engaging and educating diversity, including bolstering efficacy, highlighting the societal relevance of cybersecurity, and integrating active problem-solving exercises. We have shared our experiences, detailed designs, and activities for instructors who wish to build on our initial success. Further, we discussed our lessons learned after evaluating the modules with different demographic groups over a year-long evaluation. Ultimately, approaching cybersecurity education from a collaborative, experiential-based approach offers promise toward engaging and educating a diverse workforce.

ACKNOWLEDGEMENTS

This material is based upon work supported in whole or in part with funding from the Office of Naval Research (ONR) contract #N00014-21-1-2732. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ONR and/or any agency or entity of the United States Government.

REFERENCES

[1] ACM Committee on Computing Education. 2020. Cybersecurity Curricular Guidance. http://ccecc.acm.org/files/publications/Cyber2yr2020.pdf.

[2] Air Force Association. 2022. Cyber Patriot XV National Youth Cyber Defense Competition Registration Report 2023-2023. https://www.uscyberpatriot.org/Documents/Fact%20Sheets/CP15%20Registration%20Report%202022-2023.pdf

[3] Omer Akgul, Taha Eghtesad, Amit Elazari, Omprakash Gnawali, Jens Grossklags, Daniel Votipka, and Aron Laszka. 2020. The hackers' viewpoint: Exploring challenges and benefits of bug-bounty programs. In *Proceedings of the 2020 Workshop on Security Information Workers, ser. WSIW*, Vol. 20. Usenix, Virtual Event, 7 pages.

[4] John Aycock, Andrew Groeneveldt, Hayden Kroepfl, and Tara Copplestone. 2018. Exercises for teaching reverse engineering. In *Conference on Innovation and Technology in Computer Science Education*. ACM, Larnaca Cyprus, 188–193.

[5] César Morillas Barrio, Mario Muñoz-Organero, and Joaquín Sánchez Soriano. 2015. Can gamification improve the benefits of student response systems in learning? An experimental study. *IEEE Transactions on Emerging Topics in Computing* 4.3 (2015), 429–438.

[6] Raymond W Blaine, Jean RS Blair, Christa M Chewar, Rob Harrison, James J Raftery Jr, and Edward Sobiesk. 2021. Creating a Multifarious Cyber Science Major. In *Technical Symposium on Computer Science Education (SIGCSE)*. ACM, Virtual Event, 1205–1211.

[7] Barry W. Boehm. 1988. A spiral model of software development and enhancement. *Computer* 21, 5 (1988), 61–72.

[8] Sergey Bratus. 2007. Hacker curriculum: How hackers learn networking. *IEEE Distributed Systems Online* 8, 10 (2007), 2–2.

[9] Lorenz Breidenbach, Phil Daian, Florian Tramèr, and Ari Juels. 2018. Enter the Hydra: Towards Principled Bug Bounties and Exploit-Resistant Smart Contracts. In *27th USENIX Security Symposium (USENIX Security 18)*. USENIX Association, Baltimore, MD, 1335–1352.

[10] David Bruley. 2018. How the Best Hackers Learn Their Craft. https://www.youtube.com/watch?v=6vj96QetfTg.

[11] Tanner J Burns, Samuel C Rios, Thomas K Jordan, Qijun Gu, and Trevor Underwood. 2017. Analysis and Exercises for Engaging Beginners in Online CTF Competitions for Security Education. In *2017 USENIX Workshop on Advances in Security Education (ASE 17)*. USENIX, Vancouver, BC, Canada, 9 pages.

[12] Peter Chapman, Jonathan Burket, and David Brumley. 2014. {PicoCTF}: A {Game-Based} Computer Security Competition for High School Students. In *Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*. USENIX, San Diego, CA, 10 pages.

[13] Tom Chothia and Chris Novakovic. 2015. An offline capture the flag-style virtual machine and an assessment of its value for cybersecurity education. In *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*. USENIX, Washington, D.C, 8 pages.

[14] Thomas Cook, Gregory Conti, and David Raymond. 2012. When good Ninjas turn bad: Preventing your students from becoming the threat. *Colloquium for Information System Security Education* 16 (2012), 61–67.

[15] Seth T Hamman, Kenneth M Hopkinson, Ruth L Markham, Andrew M Chaplik, and Gabrielle E Metzler. 2017. Teaching game theory to improve adversarial thinking in cybersecurity students. *IEEE Transactions on Education* 60.3 (2017), 205–211.

[16] Maurice Hendrix, Ali Al-Sherbaz, and Bloom Victoria. 2016. Game based cyber security training: are serious games suitable for cyber security training? *International Journal of Serious Games* 3.1 (2016), 53–61.

[17] Sylvia Hurtado, Nolan L Cabrera, Monica H Lin, Lucy Arellano, and Lorelle L Espinosa. 2009. Diversifying science: Underrepresented student experiences in structured research programs. *Research in Higher Education* 50 (2009), 189–214.

[18] Blake Janes, Heather Crawford, and TJ OConnor. 2020. Never Ending Story: Authentication and Access Control Design Flaws in Shared IoT Devices. In *IEEE Security and Privacy SafeThings Workshop (SafeThings)*. IEEE, Virtual Event.

[19] Ralph Langner. 2011. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy* 9, 3 (2011), 49–51.

[20] Kees Leune and Salvatore J Petrilli Jr. 2017. Using capture-the-flag to enhance the effectiveness of cybersecurity education. In *Proceedings of the 18th Annual Conference on Information Technology Education*. ACM, Bologna, Italy, 47–52.

[21] Tzu-Chiang Lin, Ying-Shao Hsu, Shu-Sheng Lin, Maio-Li Changlai, Kun-Yuan Yang, and Ting-Ling Lai. 2012. A review of empirical evidence on scaffolding for science education. *International Journal of Science and Mathematics Education* 10 (2012), 437–455.

[22] Henry Lowood and Raiford Guins. 2016. *Debugging game history: A critical lexicon*. MIT Press.

[23] Donatello Luna, Luca Allodi, and Marco Cremonini. 2019. Productivity and patterns of activity in bug bounty programs: Analysis of HackerOne and Google vulnerability research. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*. ACM, Virtual Event, 1–10.

[24] Xenia Mountrouidou, David Vosen, Chadi Kari, Mohammad Q Azhar, Sajal Bhatia, Greg Gagne, Joseph Maguire, Liviana Tudor, and Timothy T Yuen. 2019. Securing the human: a review of literature on broadening diversity in cybersecurity education. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*. ACM, Aberdeen, UK, 157–176.

[25] NSA. 2022. Academic Requirements for Designation as a CAE in Cyber Operations Fundamental. https://www.nsa.gov/Resources/Students-Educators/centers-academic-excellence/cae-co-fundamental/requirements/

[26] TJ OConnor. 2022. HELO DarkSide: Breaking Free From Katas and Embracing the Adversarial Mindset in Cybersecurity Education. In *Technical Symposium on Computer Science Education (SIGCSE)*. ACM, Providence, RI, 710–716.

[27] TJ OConnor, Dane Brown, Jasmine Jackson, Suzaana Schmeelk, and Bryson Payne. 2023. Compete to Learn: Toward Cybersecurity As A Sport. In *Journal of Cybersecurity Education, Research and Practice (JCERP)*. Kennesaw State University.

[28] TJ OConnor, William Enck, and Bradley Reaves. 2019. Blinded and Confused: Uncovering Systemic Flaws in Device Telemetry for Smart-Home Internet of Things. In *Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*. ACM, Miami,FL.

[29] TJ OConnor, Dylan Jesse, and Daniel Camps. 2021. Through the Spyglass: Toward IoT Companion App Man-in-the-Middle Attacks. In *Cyber Security Experimentation and Test (CSET)*. USENIX, Virtual Event.

[30] TJ OConnor, Dylan Jessee, and Daniel Campos. 2023. Towards Examining The Security Cost of Inexpensive Smart Home IoT Devices. In *International Workshop on Consumer Devices, Systems, and Services (CDS 2023)*. IEEE, Torino, IT.

[31] TJ OConnor, Carl Mann, Tiffanie Petersen, Isaiah Thomas, and Chris Stricklan. 2022. Toward an Automatic Exploit Generation Competition for an Undergraduate Binary Reverse Engineering Course. In *Innovation and Technology in Computer Science Education (ITiCSE)*. ACM, Dublin, Ireland, 442–448.

[32] TJ OConnor, Alex Schmith, Chris Stricklan, Marco Carvalho, and Sneha Sudhakaran. 2024. Pwn Lessons Made Easy With Docker: Toward an Undergraduate Vulnerability Research Cybersecurity Class. In *Technical Symposium on Computer Science Education (SIGCSE TS)*. ACM, Portland, OR.

[33] TJ OConnor and Chris Stricklan. 2021. Teaching a Hands-On Mobile and Wireless Cybersecurity Course. In *Innovation and Technology in Computer Science Education (ITiCSE)*. ACM, Virtual Event, 296–302.

[34] TJ OConnor and Chris Stricklan. 2021. Towards Binary Diversified Challenges For A Hands-On Reverse Engineering Course. In *Innovation and Technology in Computer Science Education (ITiCSE)*. ACM, Virtual Event.

[35] Maria Chaparro Osman, Maureen Namukasa, Cherrise Ficke, Isabella Piasecki, TJ OConnor, and Meredith Carroll. 2023. Understanding how to diversify the cybersecurity workforce: A qualitative analysis. In *Journal of Cybersecurity Education, Research and Practice (JCERP)*. Kennesaw State University.

[36] Rodney Petersen, Danielle Santos, Matthew Smith, and Gregory Witte. 2020. Workforce Framework for Cybersecurity (NICE Framework).

[37] W Michael Petullo. 2022. Courses as Code: The Aquinas Learning System. *Proceedings of the 15th Workshop on Cyber Security Experimentation and Test* (2022).

[38] Phillip Porras, Hassen Saidi, and Vinod Yegneswaran. 2009. Conficker C analysis. *SRI International* 1 (2009), 1–1.

[39] Summer Rebensky, Maria Chaparro, and Meredith Carroll. 2020. Optimizing the Learning Experience: Examining Interactions Between the Individual Learner and the Learning Context. In *Advances in Human Factors in Training, Education, and Learning Sciences: Conference on Human Factors in Training, Education, and Learning Sciences*. Springer, AHFE, Virtual Event, 10–16.

[40] James R Rest. 1994. Background: Theory and research. *Moral development in the professions: Psychology and applied ethics* (1994), 26 pages.

[41] Wei-Cheng Milton Shen, De Liu, Radhika Santhanam, and Dorla A Evans. 2016. Gamified technology-mediated learning: The role of individual differences. In *Pacific Asia Conference on Information Systems (PACIS)*. Association For Information System, Chiayi, Taiwan.

[42] Jacob Springer and Wu-chang Feng. 2018. Teaching with angr: A Symbolic Execution Curriculum and {CTF}. In *2018 Workshop on Advances in Security Education (ASE 18)*. USENIX, Baltimore, MD, 8 pages.

[43] Kevin A Stein and Matthew H Barton. 2019. The "Easter egg" syllabus: Using hidden content to engage online and blended classroom learners. *Communication Teacher* 33, 4 (2019), 249–255.

[44] Valdemar Švábenskỳ, Jan Vykopal, and Pavel Čeleda. 2020. What Are Cybersecurity Education Papers About? A Systematic Literature Review of SIGCSE and ITiCSE Conferences. In *51st ACM Technical Symposium on Computer Science Education*. ACM, Portland, OR, 2–8.

[45] Valdemar Švábenskỳ, Jan Vykopal, Milan Cermak, and Martin Laštovička. 2018. Enhancing cybersecurity skills by creating serious games. In *23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. ACM, Larnaca Cyprus, 194–199.

[46] Clark Taylor, Pablo Arias, Jim Klopchic, Celeste Matarazzo, and Evi Dube. 2017. CTF: State-of-the-Art and Building the Next Generation. In *2017 USENIX Workshop on Advances in Security Education (ASE 17)*. USENIX, Vancouver, BC, Canada, 11 pages.

[47] Holly Tootell, Mark Freeman, and Alison Freeman. 2014. Generation alpha at the intersection of technology, play and motivation. In *2014 47th Hawaii international conference on system sciences*. IEEE, Waikoloa, HI, 82–90.

[48] Jan Vykopal, Valdemar Švábenskỳ, and Ee-Chien Chang. 2020. Benefits and pitfalls of using capture the flag games in university courses. In *Technical Symposium on Computer Science Education (SIGCSE)*. ACM, Virtual Event, 752–758.

[49] David C Webb, Alexander Repenning, and Kyu Han Koh. 2012. Toward an emergent theory of broadening participation in computer science education. In *Technical Symposium on Computer Science Education (SIGCSE)*. ACM, Raleigh, NC, 173–178.

[50] SeongIl Wi, Jaeseung Choi, and Sang Kil Cha. 2018. Git-based CTF: A Simple and Effective Approach to Organizing In-Course Attack-and-Defense Security Competition. In *2018 USENIX Workshop on Advances in Security Education (ASE 18)*. USENIX, Baltimore, MD, 9 pages.

[51] Rushan Ziatdinov and Juanee Cilliers. 2022. Generation Alpha: Understanding the next cohort of university students.