Reflections of Cybersecurity Workshop for K-12 Teachers

Chad Mourning
Harsha Chenji
Allyson Hallman-Thrasher
Ohio University
Athens, Ohio
mourning@ohio.edu
chenji@ohio.edu
hallman@ohio.edu

Savas Kaya
Nasseef Abukamail
Ohio University
Athens, Ohio
kaya@ohio.edu
abukamai@ohio.edu

David Juedes
Avinash Karanth
Ohio University
Athens, Ohio
juedes@ohio.edu
karanth@ohio.edu

ABSTRACT

In this paper, we recount efforts in developing cybersecurity workshops for K-12 teachers, intended to learn skills to better educate the cybersecurity workers of tomorrow. In 2021, we provided two one-day virtual workshops and in 2022 we provided one two-day in-person workshop to high school teachers to increase cybersecurity awareness in three areas: general cybersecurity issues, software security, and hardware security. Both the online and in-person workshops employed Google classroom and Jupyter Notebooks, and high school teachers were provided with Raspberry Pi Zeros to use as part of the workshops. This paper describes the design and implementation of the workshops and also provides evidence demonstrating the effectiveness of the workshops, as well as commentary to provide guidance for future efforts.

CCS CONCEPTS

• **Applied computing** → **Interactive learning environments**; • **Security and privacy** → **Software security engineering**; **Hardware-based security protocols**.

KEYWORDS

Cybersecurity, Assessment, High School Students, K-12 Teachers

ACM Reference Format:
Chad Mourning, Harsha Chenji, Allyson Hallman-Thrasher, Savas Kaya, Nasseef Abukamail, David Juedes, and Avinash Karanth. 2023. Reflections of Cybersecurity Workshop for K-12 Teachers. In *Proceedings of the 54th ACM Technical Symposium on Computing Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569761

1 INTRODUCTION

Ubiquitous computing and interconnected computers have led to a rapid increase in the use of these devices (smartphones, laptops, desktops, workstations) across a wide range of applications and business solutions, and it is inconceivable that these devices will ever be disconnected to improve security. Security and trust in today's computing systems have become essential at both hardware and software levels. Hardware security primitives play an important role in ensuring trust, integrity, and authenticity of integrated circuits (ICs) and electronic systems. Similarly, the software running on the hardware should also protect and defend against cyber-attacks.

A thriving workforce of digital design engineers and software cybersecurity experts, familiar with assured and trusted systems, is critically required to thwart future attacks. However, the world continues to face a growing shortage of qualified cybersecurity professionals and practitioners. It has been estimated that 1.8 million cybersecurity-related positions both in government and non-government sources will be unfulfilled by 2022 [1]. This acute shortage significantly affects businesses, industry, government agencies, educational institutions and researchers from having a response to tackle the growing threat posed by cyber-attacks. Cybersecurity is not only essential to protecting our nation's critical infrastructure, but also has become a global necessity. Therefore, there is an urgent need for developing a workforce that recognizes security vulnerabilities in computing systems and implements countermeasures to improve cyber-defenses across various devices and applications. Developing competent cybersecurity professionals must begin with K-12 teachers who are at the front lines of promoting interest in Science, Technology, Engineering, and Math (STEM) fields and preparing students with the background appropriate to succeed in STEM majors in college.

Bringing awareness to cybersecurity issues at the high-school level is a foundational step in creating a robust cybersecurity workforce. Similarly, before high school students can be prepared to take on the cybersecurity challenges of tomorrow, high school teachers, particularly in computer science, must have the skills to instruct them. In addition, fostering cybersecurity awareness in K-12 STEM teachers who do not teach computer science courses will ensure that we can bring digital awareness to all teachers and their students. As K-12 teachers awareness and knowledge of cybersecurity grows, they are better prepared to teach issues of cybersecurity to K-12 students and become more comfortable doing so.

With the goal of increasing cybersecurity awareness for teachers, in 2021, we designed and implemented multiple one-day virtual workshops addressing cybersecurity topics that included hands-on programming and hardware exercises. The workshop provided a

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569761

[1] https://www.cybersecurity-insiders.com/cybersecurity-workforce-shortage-projected-at-1-8-million-by-2022-2/

general overview of cybersecurity – what it is, why it is important, and how it looks in the real world– examples of software security issues, how to protect against cyber attacks, and an exploration of hardware security that included running programs on a Raspberry Pi [15] kit sent to participants In 2022 we were able to offer the same workshop in an in-person, two-day format.

There are many for-profit offerings of cybersecurity training available for educators seeking to improve their skills, and NIST provides a list [1] of free and low-cost cybersecurity resources for professionals, educators, and children; however, these resources either have some associated cost or are courseware with no dedicated instructors. The workshop we offered was provided at no cost to the participants, had a generous instructor to student ratio (roughly 1:3 in the virtual offering, and 7:6 in the in-person offering), and, perhaps most importantly to the participants, qualified for their continuing education requirements. During our literature review we struggled to find any other workshops that met these criteria; we are confident that such workshops or short courses must exist, but have not, necessarily, published an experience report or other scholarly work on the effectiveness of their workshop.

In this paper, we present an assessment of the effectiveness of this workshop for K-12 teachers who are computer science teachers and for teachers who are general STEM teachers, (i.e., those whose regular teaching duties do not include teaching computer science). This paper also provides detailed analysis of the experiences of 19 K-12 teachers from Ohio and can provide a blueprint for future workshops in the domain of cybersecurity. We believe that future computer science curricula must include topics in cybersecurity at an early stage of the K-12 curriculum to promote students' awareness and interest in the field of cybersecurity and, thus, ensure that we have a thriving workforce for future cyber defenses. The major contributions of this experience report are as follows:

- **Workshop Curriculum:** We describe the curriculum including the design, implementation, and topics covered in the cybersecurity workshop. We also detail some of the hands-on exercises, which were implemented via Jupyter notebooks, and hardware kits provided to attendees.
- **Assessment of the Workshop:** We share the impacts teachers report the workshop having on 1) their preparation to teach cybersecurity topics and 2) their own learning in order to provide insights into the types of professional development needed by computer science teachers and general STEM teachers to be prepared to introduce ideas of cybersecurity to their students.

The Workshop Content was closely aligned with the 2017 ACM CSEC curricula [3]. Chapter 4 of the curricula provides a comprehensive list of suggested topics which are divided into *Knowledge Areas* containing multiple *Knowledge Units* each with various *Topics*. We cover a broad spectrum of Topics, a subset of which can be found in Table 1.

Knowledge Area ***Unit*** **Topic**	**Specific Content**
Data Security *Cryptography* Basic concepts	Encryption Decryption
Data Security *Data Integrity* *and Authentication* Password Attack Techniques	Brute Force
Data Security *Cryptanalysis* Classical Attacks	Brute Force Attacks Frequency Analysis
Data Security *Access Control* Physical Security	Data Destruction
Data Security *Access Control* Logical Data Access Control	Access Control Lists
Software Security *Implementation* Validating input and checking its representation	Buffer Overflow SQL Injection
Component Security *Component Design* Component design security	Hardware Trojans Intellectual Property Piracy
Component Security *Component Design* Principles of secure component design	Chain of Trust Reducing Risk
Connection Security *Distributed Systems* *Architecture* World-wide-web	HTTPS SSL
Societal Security *Cybercrime* Cybercriminal behavior	Data Theft Identity Theft

Table 1: A Subset of Knowledge Areas, Knowledge Units, and Topics covered in the workshop from the ACM Cybersecurity Curricula. Rightmost columns are specific examples used.

2 WORKSHOP CONTENT

In the initial offering, the content was divided into three two-hour modules: a broad introduction to the theory and practice of cybersecurity, an overview of software security, and an introduction to hardware security, that included both lecture and hands-on activities. The hands-on exercises were primarily delivered via Jupyter notebooks [4], which have recently become a common platform for delivering interactive educational content [12]. The notebooks are hosted permanently online through a combination of public GitHub repositories and Binder, allowing participants access in the future for continued exploration after the workshop ended. We provided a link to mybinder.org which instantiated a separate instance of a set of Jupyter notebooks for each participant. The first-year workshops were conducted virtually via Zoom and had a high instructor to participant ratio (1:3) which allowed us to place 1-2 instructors in each zoom breakout group during the hands on exercises. Each module concluded with a quiz over content drawn from both the

lecture and the exercises. The second year offerings were conducted in-person; the lecture portions of the workshop were roughly the same in length, but extra time was given for the hands-on examples, and a few additional, popular videos were shown.

2.1 Introduction to Cybersecurity

The opening two-hour module was designed to provide an overview of cybersecurity. The main goal was to demonstrate that the study of cybersecurity is not just confined to cryptography, mathematics, and software programming, but in fact includes many diverse areas such as risk management, threat modeling, and hardware circuits. We intentionally used this broad overview to appeal to participants with a wide range of backgrounds and technical knowledge. In the initial offering, hands-on experience and demonstrations were used during 30-45 minutes of the module to support principles of active learning, wherein participants learn by engaging directly with the content themselves and have to make decisions about principles and concepts without being directly told by the workshop instructors.

The module was broadly divided into four parts that included both lecture and hands-on exercises: 1) confidentiality, integrity, and availability (C-I-A); 2) threats and threat modeling; 3) access control lists (ACLs), policies and mechanisms; and 4) assurance and trust with new content presented in 15-30 minute segments to promote continued engagement in a virtual setting. We introduced the classic model for computer security (i.e., Alice and Bob who communicate over an insecure channel) to address the concepts of C-I-A, ciphers, en(de-)cryption and brute force attacks. After discussing confidentiality, participants were divided into breakout rooms where they were actively encouraged to construct a solution collaboratively to a Python-based exercise set up in a Jupyter notebook on brute forcing a simple one time pad cipher. We used threat modeling [13] via two brief examples (the STRIDE model [13] and the MITRE ATT&CK [8] models) as a useful abstraction to reason about risks early in the design process. To discuss ACLs we used an illustrative example where the importance of the "*-property" in the Bell-LaPadula model[14] was highlighted using a simple access control matrix. The final part discussed human and operational issues as well as social engineering. We concluded with a demonstration of a capture the flag exercise [5].

2.1.1 Python exercises. The first notebook introduced elementary Python using the topic of digital representation of data where attendees could see how to convert between number systems, recognize the need for compact hexadecimal representations of data, and a standardized encoding scheme (e.g., ASCII). We also introduced the BitVector library as a example of useful methods for bit-wise manipulation of data, such as XOR. The second notebook discussed the One Time Pad cipher [7], which involves using an XOR operation to both encrypt and decrypt. Using a truth table for XOR, participants saw how the ASCII representation of a common word such as `hello` could be encrypted. Then participants completed exercises that required them to brute force ciphertext over the entire key space of the cipher in order to help them see why this cipher is considered secure. Due to the computational limitations of the mybinder instance, only a two letter word (16 bits in ASCII) was provided. Participants had to try every possible key, check whether the decrypted plaintext was a valid word by looking it up in a dictionary, and, finally, print it to output if it was valid. Both the design of the Jupyter notebook and instructor support were intended to scaffold the codewriting experience for participants (Figure 1). Teachers with no or limited programming experience had some struggles with the exercises, but during the two-day workshop, ample time was given for all participants to complete the assignment.

```
wordlist = []
for line in open("_________________"):
    if _________ <= ____:
        wordlist.append(____________)
ctext_bv = BitVector(hexstring="_____")
ctext_length = ________
key_bv = BitVector(size=__________)
for i in range(0, ______):
    key_bv.set_value(intVal=_______, size=_______)
```

Figure 1: Code snippet that participants used to construct a brute force loop for the OTP cipher. The underscores represent blanks that they had to fill in.

Table 2: Capture the Flag Attack Orchestration.

Step	Action	Result
1	Initiate port scan using `nmap`	List of open ports, including port 80 (HTTP)
2	Point web browser to port 80	View photo blog and notice use of PHP in the URL
3	Use `sqlmap` tool on the URL	Perform SQL injection; obtain the blog's admin's password in clear text
4	Login as admin user	Privilege escalation; ability to upload images
5	Upload a shell script disguised as an image	Obtain a backdoor
6	Click on uploaded image	Script executes; reverse shell becomes active
7	Execute `ls` command on the shell	Obtain a listing of files on the machine

2.1.2 Capture the Flag. After the lecture was complete, the instructor showed a demonstration of a capture the flag exercise (Table 2) instantiated on an isolated network consisting of virtual machines and a pfSense based firewall. Kali Linux was used as the attacker machine, and a readily available boot image [5] was chosen as the victim. The goal was to obtain a listing of files on the target, by exploiting vulnerabilities in the photo blog server software. This demo closed the loop by collecting most of the concepts espoused in the lecture and stitching them together in a single context.

2.2 Software Security

The two-hour software security module (again split between lectures and hands on exercises) included: Physical Security, Motivations for Cybercrime, SQL Injection, Buffer Overflow Attacks, Cross Site Scripting, and Practical Aspects of Cryptography.

The hands on exercises focused on the fundamentals of cryptography, practical attacks against passwords such as frequency

analysis [11] and SQL Injection. These exercises were interleaved into the lecture at appropriate points to create a heterogeneous experience in an attempt at maintaining participant stimulation and sustained engagement. Exercises could be completed with no prior programming experience; therefore, only minor changes in code were needed, but the scope of the output was large. Participants needed very little instructor intervention during their exercises. The following sections describe the purpose of each exercise.

2.2.1 Cryptographic Attacks. This section started with notebooks including discussions of classic encryption/decryption of a cipher/-plaintext, one time pads, modern public key encryption, and how encryption is used in modern society. This was followed by a pair of notebooks containing interactive examples for participants to perform dictionary attacks on a password and frequency analysis of a simple substitution cipher.

The notebook loads a priming text to generate the vowel and space distributions in an English language sample, loads the results of a substitution attempt, finds the vowel and space distribution of the decipher attempt, and finally runs the distance metric found in Equation 1 on the resulting distribution vectors, printing the results.

$$\sqrt{\sum_{i=1}^{6}(x_i - y_i)^2} \tag{1}$$

The distribution with the lowest difference is assumed to be English, and, therefore, the correct substitution has been applied. Participants were tasked with running the program on various substitutions and recording which generated the lowest distance. The notebook also contained advanced exercises meant for consumption by participants after the conclusion of the workshop.

2.2.2 SQL Injection. This notebook explained how a user or program might use SQL to interact with a database and demonstrated the negative consequences of unsanitized database queries. Two interactive examples allowed participants to perform SQL Injection attacks exploiting an existing database to reveal a trove of (fictional) employee social security numbers as well as edit salary information to give themselves a raise. The goal of these hands-on exercises was to provide awareness to the participants on how cyber-attacks could be implemented and who can be trusted to perform sensitive operations. Moreover, by providing detailed discussion on mitigation techniques, the participants were aware of steps that could, and should, be taken to protect from such attacks.

2.3 Hardware Security

The final two-hour module of the workshop focused on hardware security with a detailed discussion on the central issues, attack types, and possible countermeasures in a manner accessible to non-specialists. The review started with a general description of design abstraction layers and industry trends such as *system-on-chip* integration and proliferation of *fabless* companies that are behind the increasing vulnerabilities in terms of Intellectual Property (IP) security, hardware trojans, and piracy [2]. The module also described supply-chain, physical and side-channel attacks, as well as necessity of establishing root-of-trust hardware in a given design. Internet-of-Things (IoT) security and portable electronics with wireless interfaces have been given a special focus in the session as they increase the attack surfaces immeasurably. The attendees were given examples like PCB and chip probing, DRAM RowHammer [9], or ThunderClap [6] attacks. Additional training resources, links for web-portals and educational videos were also provided to the attendees to raise awareness on hardware security and indicate existing materials helpful for course development and career planning.

Hardware Trojan. A Jupyter notebook was developed to be used directly on the Raspberry Pi kits sent to participants. In addition to the Raspberry Pi, the kit included a Pimoroni Enviro equipped with light, temperature, proximity, and audio sensors [10]. The example included a benign looking application that polled the sensors and printed the output, but it included two hardware interfaced backdoors that the participants could trigger.

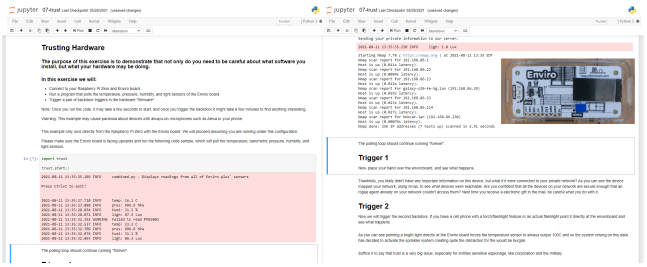

Figure 2: Left) The trusted application operation under normal conditions. Right) The application when applying Trigger 1. Notice it discovered seven devices, some named, on the network. The inset shows the Rpi0 with Enviro board.

The first backdoor example demonstrated the importance of running certified software on IoT devices on your home network. In this case, if a participant placed their hand in close proximity to the board, the application would run an `nmap` scan of their home network, listing all connected devices and, fictionally, sending this information to a remote server. An example of the output before and after the first backdoor trigger can be found in Figure 2.

The second backdoor was a contrived example demonstrating how a third party might illicitly use a hardware backdoor to change the operational state of a network via an attached device. In this example, the imagined network's sprinkler system and fire alarms are intended to trigger when the Envrio temperature sensor reaches 100° C. If a participant shined a bright light source (> 300 lux), such as a cell phone flashlight, at the *Enviro* light sensor, then the software would intercept the temperature reading and replace it with 100° C, triggering the fictional sprinkler system and allowing the unauthorized entity a few minutes alone in the building until the fire department arrived.

3 EFFECTIVENESS OF WORKSHOP

Aggregate pre- and post-survey responses related to knowledge, confidence, and future teaching can be found in Figure 3. The primary goal of the workshop of the workshop was improving teachers' knowledge regarding cybersecurity, increasing awareness of cybersecurity issues, and developing teachers' confidence to introduce these topics to students. To determine how effective the workshop was at accomplishing these goals, we used a pre- and

	Pre-Workshop Survey						Post-Workshop Survey					
	I am confident in **my own knowledge of computer science.**	I am confident in **my ability to teach computer science topics.**	I am confident in **my own knowledge of programming.**	I am confident in **my ability to teach programming.**	I am confident in **my own knowledge of cyber security.**	I am confident in **my ability to teach topics related to cyber security.**	I am more **confident** in my own **knowledge of cyber security.**	**I am confident to introduce students to cyber security** topics next year.	I am more **confident** in my ability to **teach programming.**	**I learned new resources and materials** for teaching computer science.	I have **learned new ideas to use in my own teaching** next year.	I anticipate **using topics from this workshop in my teaching** next year.
Overall Teacher Average (n = 19)	3.158	3.105	3.263	3.316	2.579	2.421	3.789	3.158	3	4.053	3.789	3.421
CS Teacher Average (n = 12)	3.5	3.583	3.583	3.667	2.75	2.833	3.917	3.333	3.091	4.083	3.833	3.833
Non-CS Teacher Average (n = 7)	2.571	2.286	2.714	2.714	2.286	1.714	3.571	2.857	2.857	4	3.714	2.714
Average Cohort 1 (n = 14)	3.214	3.142	3.285	3.357	2.643	2.571	3.643	3.214	2.923	4	3.786	3.429
Average Cohort 2 (n = 5)	3	3	3.2	3.2	2.4	2	4.2	3	3.2	4.2	3.8	3.4

Figure 3: Teacher Likert-Scale Responses Related to Knowledge, Confidence, and Future Teaching.

post-survey which included both open-ended questions and statements to which participants could respond with a five-point Likert scale (5 = strongly agree to 1 = strongly disagree). The pre-survey collected information on teacher backgrounds, content preparation, teaching experience, and goals for the workshop. The post-survey asked participants to describe cybersecurity and its importance, give general feedback on the workshop, and what they learned or found difficult. Additionally, teachers were asked to evaluate the workshop's content for its applicability to their classes/students. Of the 22 teachers 19 completed both the pre- and post-survey and these are the participants whose data is reported here. We share results of both computer science teachers (n=12) and general STEM (n=7) teachers who do not teach computer science, hereafter referred to as CS teachers and non-CS teachers, respectively.

3.1 Impact on Preparation to Teach Cybersecurity

According to pre-survey results, CS teachers were confident in their content knowledge in areas of general computer science, programming, and cybersecurity. Responses were slightly higher in the first year's cohort than the second. It is important to note that cybersecurity was the area in which CS teachers were least confident going into the workshop. CS teachers were similarly confident in teaching computer science and programming, but less confident with their ability to teach cybersecurity. Non-CS teachers, on the other hand, were far less confident in their content knowledge than CS teachers. While unsurprising that content knowledge and teaching confidence for computer science content is low among teachers for whom computer science is not a typical teaching responsibility, we share this finding for the purpose of noting theses teachers' growth in these areas after the workshop.

In the post-survey, all but 3 (15%) CS teachers reported increased confidence in their content knowledge of cybersecurity and 14 of 19 (74%) reported increased confidence in teaching cybersecurity.

Further of the 12 CS teachers, 11 "agreed" or "strongly agreed" that they had learned new materials for teaching computer science, 9 noted that they learned ideas that they can be used in their teaching for the next year, and 8 informed that they anticipate using ideas from the workshop in their teaching in the coming year. Specifically, 3 CS teachers plan to incorporate Jupyter notebook activities, a fourth wants to address Python coding more meaningfully, and two others were interested in using other resources that had been shared.

All but two non-CS teachers reported increased confidence in their content knowledge of cybersecurity and 6 of the 7 non-CS teachers reported increased confidence in teaching cybersecurity. The average Likert-scale rating of non-CS teachers increased from 2.286 to 3.571 in terms of cybersecurity knowledge confidence and increased from 1.714 to 2.857 in terms of confidence in teaching cybersecurity. Interestingly, 6 of the 7 non-CS teachers noted that they might use material from the workshop in their teaching. Two made connections to their content through the Jupyter notebook activities; one noted that it could help students better understand the importance of password security which is an on-going problem his students experience, and two others noted connections between programming, cryptography, and mathematics. Another noted an interest in incorporating a hardware project and several mentioned the importance of helping students understand what cybersecurity was and how it affected them.

We claim that both CS and non-CS teachers' cybersecurity awareness has increased due to the topics covered in the workshop. This awareness could positively impact the teachers' ability to share their ideas on cybersecurity with their own students. We are excited about the improvements for those teachers not teaching computer science because part of the premise of our workshop is that cybersecurity effects everyone and we want to increase awareness for students beyond those who might typically be enrolled in a computer science class. One concern was that across both groups of teachers' self-efficacy for teaching programming seemed to decline. We might expect teachers' feeling of competence with programming to be negatively influenced by the workshop if the programming required for the workshop activities was beyond the

	I had adequate programming knowledge to engage in the workshops activities.	The lecture topics were interesting.	The problem sets were interesting.	The problem sets were appropriately challenging to me.	The problem sets were appropriately challenging for [illegible]	Lecture topics were presented at an appropriate level of rigor for me.	Lecture topics were presented at an appropriate level of rigor for my students.	I learned something new about cyber security.	I learned something about the importance of cyber security.	I learned something from the software security session	I learned something from the hardware security session.	Overall, this workshop met my expectations.
Overall Teacher Average ($n = 19$)	3.158	3.684	3.778	3.053	2.526	3.368	2.632	4.421	4.368	4.368	3.895	3.368
CS Teacher Average ($n = 12$)	3.25	3.75	3.818	2.833	2.667	3.25	2.75	4.167	4.083	4.25	3.917	3.5
Non-CS Teacher Average ($n = 7$)	3	3.571	3.714	3.429	2.286	3.571	2.429	4.857	4.857	4.571	3.857	3.143
Average Cohort 1 ($n = 14$)	3.286	3.571	3.846	2.929	2.5	3.643	2.786	4.429	4.357	4.429	3.786	3.429
Average Cohort 2 ($n = 5$)	2.8	4	3.6	3.4	2.6	2.6	2.2	4.4	4.4	4.2	4.2	3.2

Figure 4: Teachers' Learning.

teachers' skill sets, as was the case with some of the non-CS teachers. However, this result held true for the CS teachers as well, many of whom were adept with programming and teaching programming. We suspect that more support is needed for teachers to engage in the Jupyter notebook activities using Python and that this finding may be due in part to the 2021 cohort who received help with troubleshooting their programs in a virtual environment. Because only one teacher could screen share at time, it was challenging to support all teachers equally in the same way that an instructor could with in-person teaching where a quick glance around the room shows if students are making progress or needs assistance. In the virtual environment programming activities were more productive when workshop attendees took turns sharing their screens to show Python code and workshop participants collaborated to help each other troubleshoot programming. The 2022 in-person cohort such collaboration was commonplace, and these teachers reported no decline in their confidence to teach programming.

3.2 Impact on Teacher Learning

Figure 4 contains survey results about teacher learning. Our analysis of workshop feedback indicates that the workshop was generally successfully in achieving its goals. Both CS and non-CS teachers found the topics and problem sets equally interesting and that lecture topics and hands-on problem sets were appropriately rigorous and challenging, despite non-CS teachers self-reporting having lesser background to engage in workshop activities. Nearly all CS and non-CS teachers reported agreement or strong agreement that they learned something new about cybersecurity, software security, hardware, and the importance of cybersecurity. We have taken note that the module on hardware security was less informative than the other two modules in the virtual setting. We suspect that this was due in part to place in the sequence of activities, at the end of a long day, and in part to the complexity of working through hands-on hardware activities in a virtual setting. However, the in-person hardware session was very successful (average 4.2 Likert scale response from the 2022 cohort) with one enthusiastically describing in class this was the best session of the workshop. While teachers found the content appropriate for themselves as learners, they disagreed that the content was appropriately challenging and rigorous for their students. From this finding we have learned that future workshops need to devote more attention to how to make this content accessible to middle and high school students.

4 CONCLUSIONS

With the goal of increasing cybersecurity awareness for teachers, we designed and implemented multiple one-day virtual workshops and an in-person two day workshop addressing cybersecurity topics that included hands-on programming and hardware exercises. Our data show that the workshop increased awareness among K-12 teachers, who teach computer science and those who do not. Nearly all CS and non-CS teachers reported agreement or strong agreement that they learned something new about cybersecurity, software security, hardware, and the importance of cyber security. We learned much about how to offer an effective virtual workshop on cybersecurity: hands-on activities were productive for fostering interest and engagement, but in a virtual setting need extra scaffolding and teachers need support in making content accessible and challenging to students. In our in-person offering of the workshop programming and hands-on exercises were able to be collaborative with immediate feedback easily provided to everyone. In-person teachers reported a better experience in the hardware portion of the workshop especially. We continue to work at developing strategies for making the workshop content accessible to all skill levels.

ACKNOWLEDGMENTS

This work was supported by grants from Ohio Department of Higher Education, Air Force Research Lab FA8650-20-2-1136 and National Science Foundation CCF-1936794.

REFERENCES

[1] 2022. https://www.nist.gov/itl/applied-cybersecurity/nice/resources/online-learning-content

[2] Swarup Bhunia and Mark Tehranipoor. 2018. *Hardware security: a hands-on learning approach.* Morgan Kaufmann.

[3] Cybersecurity Curricula. 2017. Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity. A Report in the Computing Curricula Series Joint Task Force on Cybersecurity Education. ACM, IEEE, AIS, IFIP, USA (2017).

[4] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. 2016. *Jupyter Notebooks-a publishing format for reproducible computational workflows.* Vol. 2016.

[5] Pentester Lab. 2021. Pentester Lab: From SQL injection to Shell. Retrieved 2021-08-11 from https://www.vulnhub.com/entry/pentester-lab-from-sql-injection-to-shell,80/

[6] Theo Markettos, Colin Rothwell, Brett F Gutstein, Allison Pearce, Peter G Neumann, Simon Moore, and Robert Watson. 2019. Thunderclap: Exploring vulnerabilities in operating system IOMMU protection via DMA from untrustworthy peripherals. (2019).

[7] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. 1996. *Handbook of Applied Cryptography* (1st edition ed.). CRC Press, Boca Raton.

[8] MITRE. 2021. MITRE ATT&CK®. Retrieved 2021-08-11 from https://attack.mitre.org/

[9] Onur Mutlu and Jeremie S Kim. 2019. Rowhammer: A retrospective. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 8 (2019), 1555–1571.

[10] Pimoroni. 2021. Enviro Plus. https://learn.pimoroni.com/tutorial/sandyj/getting-started-with-enviro-plus

[11] Edgar Allen Poe. 1843. The Gold Bug. *Dollar Newspaper* (1843).

[12] Bernadette M Randles, Irene V Pasquetto, Milena S Golshan, and Christine L Borgman. 2017. Using the Jupyter notebook as a tool for open science: An empirical study. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL).* IEEE, 1–2.

[13] Adam Shostack. 2014. *Threat Modeling: Designing for Security* (1st edition ed.). Wiley, Indianapolis, IN.

[14] William Stallings and Lawrie Brown. 2017. *Computer Security: Principles and Practice* (4th edition ed.). Pearson, New York, NY.

[15] Eben Upton and Gareth Halfacree. 2014. *Raspberry Pi user guide.* John Wiley & Sons.