Teaching a Hands-On Mobile and Wireless Cybersecurity Course

TJ OConnor, Christopher Stricklan
{toconnor,cstricklan}@fit.edu
Florida Institute of Technology
Melbourne, FL, USA

ABSTRACT

The combination of theory-based and practical hands-on learning represents a powerful approach for cybersecurity education. Placing the student in the adversarial mindset strengthens this approach and is commonly exercised in network penetration testing, reverse engineering, and binary exploitation coursework. In this paper, we present an undergraduate mobile and wireless security course design that balances theoretical learning with a hands-on and adversarial thinking approach. Our course consists of inter-woven lectures and lab sessions. Labs consist of contemporary attacks against radio-frequency (RF) enabled hardware, Internet of Things (IoT) firmware, and wireless protocols. In the culmination exercise, the students attack a flawed RF protocol implemented on GnuRadio to allow students to demonstrate their knowledge synthesis. We believe that sharing this experience will prove valuable for instructors who wish to introduce adversarial thinking into mobile and wireless security courses while overcoming the challenge of remote students.

CCS CONCEPTS

• **Social and professional topics** → **Model curricula**; **Computing education programs**; • **Networks** → **Mobile and wireless security**.

ACM Reference Format:
TJ OConnor, Christopher Stricklan. 2021. Teaching a Hands-On Mobile and Wireless Cybersecurity Course. In *26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2021), June 26–July 1, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3430665.3456346

1 INTRODUCTION

Several recent high-profile attacks have exploited design and implementation flaws in wireless and mobile protocols. Notable examples include a clone attack against the Tesla Bluetooth Key Fob, an iPhone radio-proximity exploit, a Ring Doorbell cryptographic design flaw, and an eavesdropping attack against the LTE cellular protocol [6, 27, 29, 39]. Mobile and wireless connectivity has become omnipresent in our lives, integrating everything from our homes, automobiles, workplaces, and tracking right down to our fitness habits over a variety of RF protocols. As we begin integrating Internet-of-Things (IoT) into our homes and Industrial IoT (IIoT) into our workplaces, there is a greater demand for a workforce that can design, implement, test, and verify secure wireless and mobile protocols.

However, a survey of cybersecurity education papers noted that cybersecurity courses predominately focus on secure programming, network security monitoring, ethical hacking, human aspects, cryptography, and authentication [45]. While each of these topics contributes to understanding mobile and wireless cybersecurity, few university courses explore mobile and wireless cybersecurity as an independent course. We hypothesize that this may be due to continually updating standards, novel attack vectors, and hardware demands that complicate the study of mobile and wireless cybersecurity. Ultimately, instructors do not want to be the aging professor teaching WEP RC4 flaws as a novel attack vector in 2020, 16 years after the cryptographic method was deprecated.

We hypothesize that leveraging adversarial thinking presents a solution to the unique challenges of implementing a hands-on mobile and wireless cybersecurity course. Modern adversarial thinking or *red-teaming* traces its military roots to General Van Riper, a military strategist who succeeded with an unconventional strategy against technologically superior military forces in a $250 million dollar war game [18]. Hamman et al. describe *adversarial thinking* as the ability to engage in strategic reasoning and present approaches for developing this mindset in the classroom [19]. Embracing this mindset has shown promise in achieving educational outcomes for network security and ethical hacking courses [46]. Adversary thinking synthesizes an understanding of technology capabilities, embracing unconventional perspectives and the ability to reason vulnerabilities.

In this paper, we share a detailed description of the design, course modules, and hands-on labs for an undergraduate mobile and wireless cybersecurity course with an emphasis on developing an adversarial mindset. Further, we explore the challenges of implementing this curriculum via remote learning due to the constraints of the recent novel coronavirus. This paper makes the following contributions:

(1) We share our experiences, lessons learned, and materials for embracing an adversary thinking focused course on mobile and wireless cybersecurity.
(2) We discuss the challenges of remote learning on experiential and cooperative learning methods and our experience overcoming these challenges in a mobile and wireless cybersecurity course.

Organization: Section 2 investigates previous work related to gamification and practice-based wireless coursework. In Section 3, we provide an overview of our course and the design approach. Section

ITiCSE 2021, June 26–July 1, 2021, Virtual Event, Germany

ACM ISBN 978-1-4503-8214-4/21/06…$15.00
https://doi.org/10.1145/3430665.3456346

4 examines the course labs. Section 5 offers insight and examines future challenges. Section 6 summarizes our conclusions.

2 PRIOR WORK

Previous works have explored the benefits of gamification in cybersecurity education [11, 13, 22, 47, 52] Švábenskỳ et al. discussed games for creating an adversarial mindset in an ethical hacking focused cybersecurity course [46]. While mobile, cellular, and wireless Security are NSA CAE Cyber Operations academic requirements [33], these subjects have not been deeply explored using a gamification approach. A review of 1,748 papers from SIGCSE and ITiCSE identified reverse engineering, network security monitoring, and ethical hacking as the most routinely addressed topics [45]. Further, educators have used classroom capture-the-flag (CTF) competitions to encourage greater cooperation and interest in cybersecurity topics [11]. Burns et al. analyzed 3,600 security challenges from CTF competitions, identifying a focus on cryptography, penetration testing of web vulnerabilities, reverse engineering, forensic analysis, and binary exploitation [10].

Arguably, instructors do not embrace gamification for mobile and wireless cybersecurity to the same degree as these topics. Continually updating mobile and wireless protocol standards, novel attack vectors, hardware demands and the complexity of constructing lab environments complicate the difficult of creating gamified education environments for mobile and wireless cybersecurity. Open source platforms and SDRs offer a promising approach for hands-on mobile and wireless cybersecurity education. Bazdresch describes classroom experiments for SDRs [5]. Sahin et al. describes a hands-on graduate course *lectures with labs* approach that encouraged students to implement various attack mitigation techniques on SDRs [41].

Outside academia, wireless CTFs exist as the DEFCON, and ShmooCon hacker conferences have hosted multiple wireless CTF competitions. The DEFCON competition focused on static skill development investigating signal processing [20]. The ShmooCon competition provided contestants with a dynamic competition forcing competitors to reverse engineer and synthesize the signal for a wireless shock collar [23]. Both CTFs leveraged software-defined radios and the open-source GnuRadio platform.

3 COURSE OVERVIEW AND DESIGN

This section describes the course model for our undergraduate mobile and wireless cybersecurity course. The course meets twice a week for 75 minutes sessions over 15 weeks. It is the third course in a sequence of six courses created to meet the outcomes of the CAE Cyber Operations Knowledge Units [33]. The course addresses the specific CAE Knowledge Units *M5: Cellular & Mobile Technologies* and *O2: Wireless Security*. Based on these standards and the course design, student outcomes include the following:

(1) Students will be able to perform signal processing to be able to analyze a signal of unknown origin, reverse engineer the protocol, and synthesize a signal for replaying.
(2) Students will be able to describe and demonstrate the vulnerabilities with ineffective mechanisms for encryption and authentication in RF-based protocols, including WiFi, Bluetooth, ZigBee, Zwave, LoRa, GSM, and LTE.
(3) Students will be able to describe the interaction of elements within a mobile architecture and end-to-end delivery of a signal between different generations of cellular and mobile network technologies.

Table 1: The course balances theory-based lessons with practical labs and contemporary research.

Lesson	Supporting Lab	References
RF Signal Processing	Lab 1	[35]
RF Protocol Analysis	Lab 1	[37, 42]
WiFi MAC Layer	Lab 2	[15, 34]
WiFi Cryptography	Lab 3	[48–51]
IoT Vulnerabilities	Lab 4	[7, 12, 21]
ZigBee Protocol	Lab 5	[31, 54]
Zwave Protocol	Lab 5	[3, 17]
Bluetooth Protocol	Lab 5	[1, 24, 44]
GSM Protocol	Instructor demo	[9, 32, 36, 38]
LTE Protocol	Instructor demo	[8, 30, 39, 40]

Apart from meeting the NSA CAE Cyber Operations outcomes, we grounded the course in pedagogical theory and teaching practice. We focused these efforts on increasing adversarial thinking through gamification, experiential, and collaborative learning. As this is the third course in the sequence, previous courses have primed students for solving problems with an unconventional approach in an adversarial mindset.

In Section 4.1, we describe engaging students in semester-long CTF game. We balance gamification and a cooperative learning environment due to our concerns about the impact of competitive environments on historically marginalized populations in cybersecurity education. With respect to instructional strategies, although males have shown to learn best in competitive environments with clear winners, females are more receptive to collaborative learning environments as they tend to be more mastery-oriented [4, 43]. Further, belongingness has been key for emotional and behavioral engagement across multiple institutions in STEM [53].

Finally, we embrace experiential learning by leveraging IoT devices to illustrate aspects of the curriculum (e.g., WiFi, Bluetooth, ZigBee, and Zwave protocols). We selected this on the finding of Chothia et al., who found that the current low level of IoT device security lends itself well to cybersecurity education since basic vulnerabilities are easy to find with few prerequisite skills, increasing student satisfaction [13].

The course topics, listed in Table 1, increase in difficulty building from an analysis of processing simple on-off-keying RF signals to attacking complex cellular protocol designs. To support each lesson, we discuss both theory and contemporary attacks listed referenced in the table. We follow a *lecture with labs* model where each theory-based lesson corresponds to a practical lab. To avoid illegal transmission of GSM or LTE cellular traffic that would interfere with other university students, these labs are illustrated by a series of instructor demonstrations during course lectures against a GSM base station under the instructor's control.

4 COURSE IMPLEMENTATION

This section describes the course implementation, including the semester-long CTF competition, the course labs, and necessary infrastructure. As the course occurred during the novel coronavirus time, the following section also addresses the methods for implementing remote learning.

4.1 Semester-Long CTF Competition

In place of static homework assignments, we organized a semester-long CTF competition to reinforce the course topics. The use of capture-the-flag events has offered promise in creating learner enthusiasm and involvement in binary reverse engineering, exploit development, and network penetration testing topic areas [2, 47, 52]. Based on the simple portal, ease of configuration, and automated system scoring engine - we implemented our CTF on the CTFd framework [14]. The instructor released five challenges per week, with point values from 100 to 500 points. The difficulty of challenges ranged from static analysis of a signal to dynamic fuzzing of an emulated device. To remotely offer dynamic challenges, we constructed Docker [25] containers to emulate several different device types. The purpose of these more dynamic challenges was to force students to embrace unconventional perspectives. For example, the 500 point Bluetooth challenge contained a buffer over-read vulnerability in the spirit of the HeartBleed [26]. Students succeeded by forging Bluetooth frames an incorrect header length in L2CAP Echo Requests. Coincidentally, we released this challenge about the same time as CVE-2020-24490 [28] (a Bluetooth vulnerability in Android Kernel that stemmed from a buffer overflow in Bluetooth 5 advertisement frames.)

Figure 1 depicts the solve rates for each of the challenges. The first 100% solved challenge (Lab-Intro-100) asked students to open a GnuRadio file sink and measure signal amplitude. The two challenges with the least solve rate (31.5%) challenged students to craft ZigBee and Zwave frames to exploit simulated devices. A quarter of the students completed 100% of the challenges. Although not assigned homework, we required students to give their best effort towards challenges by using their CTF score to determine a class participation grade (worth 10% of the overall course.) This approach helped to balance intrinsically motivated students who found the competition to be gimmicky. To avoid added pressure of the scoreboard, we explained that students who scored three challenges or more weekly would receive outstanding marks.

4.2 Course Labs

The five course labs were conducted in a cooperative learning environment with instructor assigned groups. Each lab focused on building student confidence and understanding of course topics by embracing an adversarial thinking approach. In contrast, the final lab provided an individual challenge for course concepts by attacking a novel protocol.

4.2.1 Lab 1: Rapid Radio Reversing. This lab introduced students to the concept of reverse engineering RF signals using software-defined radios. The instructor shipped remote students a software-defined radio and instructed them to study a remote control device in their environment by reverse-engineering the device's RF signal. Students selected various devices, including car key fobs, garage door openers, doorbells, power outlets, and remote-controlled cars. First, the students researched the device's frequency by looking up the FCC database's device identifier. They then used a software-defined radio and Universal Radio Hacker [37] software suite to identify the modulation scheme. Finally, students constructed a GnuRadio flow-graph to encode and synthesize a signal. This lab helped to build enthusiasm for remote instruction as students shared Snapchat videos of themselves analyzing the devices in their homes as depicted in Figure 2.

4.2.2 Lab 2: Wireless Denial-of-Service. This lab covered the basics of understanding 802.11 control, management, and data frames. The instructor divided the students into four teams, giving each group a Raspberry Pi connected to a WiFi access point (AP). The instructor then showed the lab scoreboard, which recorded points from the Raspberry Pi remained connected to the AP. Next, the instructor demonstrated how the scoreboard stopped scoring points for an individual team when they disconnected their Raspberry Pi. The instructor explained the goal: to have the highest score at the end of the lab. Students remotely interacted with the over a virtual private network (VPN) tunnel. Once connected to the VPN, students could log in to a lab machine with a WiFi radio capable of monitoring and spoofing 802.11 frames. Many paths existed to success, including a coarse-grained approach of forging malicious control traffic (e.g., CTS and RTS floods.) Additionally, students could apply a more fine-grained method of crafting unencrypted management traffic (e.g., de-authentication frames and false beacons.) However, the real key to success became the ability to score more points by eavesdropping and replaying the unencrypted data frame, which contained a key that each Raspberry Pi periodically sent to the scoreboard server. By analyzing the lab vulnerabilities instead of just applying attack techniques, these students realized they could substantially outscore their opponents.

4.2.3 Lab 3: Wireless Intrusion Detection System. After exploring different contemporary methodologies for attacking WiFi networks' cryptographic schemes [48–50], we shifted the paradigm and asked students to become the network defenders. In this lab, students wrote an intrusion detection system to identify an attacker on the WiFi network. The instructor gave students sample code for passively eavesdropping on the wireless medium and parsing 802.11 frame headers. The students must transform this code into a working wireless intrusion detection system. For example, a snippet of student code in Listing 1 identifies when de-authentication frames are spoofed to force a WPA2 4-way handshake to occur (as a precursor to a brute-force attack that calculates the pairwise transient key.) The only real path to success in this lab is a strong understanding of the methodology an attacker uses to compromise WiFi cryptography. For example, several teams analyzed the source code of the aircrack-ng to understand how it implemented various attack techniques. This lab helped reinforce a course theme that an adversary mindset can empower a strong defense.

4.2.4 Lab 4: IoT Firmware Backdoor Creation. This lab familiarized students with the methods of IoT firmware analysis. Students learned techniques for unpacking IoT firmware, mounting compressed filesystems, and analyzing firmware forensic artifacts in the

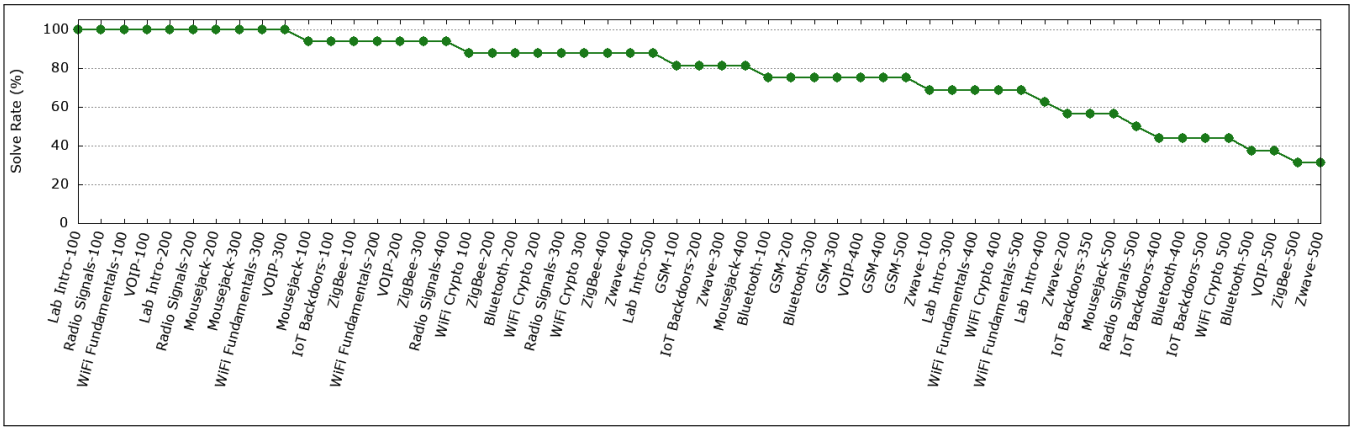

Figure 1: Our course included a remotely available, semester-long capture-the-flag competition, which included problems with varying degree of difficulty.

Figure 2: We observed students' enthusiasm rose as they shared Snapchat videos of reverse-engineering RF signals in their home environments.

Listing 1: Embracing an adversary mindset simplified the process of detecting attacks during the WiFi IDS lab.

```
if pkt.type == TYPE_MGT:
    if pkt.subtype == 12:
        print(f"Deauthentication Frame\n"
              f"[802.11 Data: Subtype:{pkt.subtype}"
              f"\tDest:{pkt.addr1}"
              f"\n\tSender: {pkt.addr2}"
              f"\n\tBSSID: {pkt.addr3}")
```

previous lesson. To demonstrate this methodology, the instructor unpacked and analyzed several firmware samples containing malicious backdoors. At the lesson's culminating point, the instructor unpacked a benign firmware sample for a camera system, added a backdoor account, repacked, and flashed the backdoored firmware sample onto an existing camera system. During the lab, the instructor split students into teams and tasked each group to find and download firmware samples on vendor support sites. The instructor challenged students to modify the downloaded samples to include a malicious backdoor to replicate a software supply chain attack. Students choose different types of devices, including doorbells, camera systems, and video surveillance recorders. The instructor explained that the other teams would gain access to their samples and receive extra points to identify the modified backdoor following the lab. The teams turned in several creative approaches for concealing their backdoors. These approaches included adding remote service accounts, inserting arbitrary command injection into web processes, hijacking benign device processes for C2 channels, and constructing remote access toolkits activated by port-knocking.

4.2.5 Lab 5: IoT RF Protocol Investigation. This lab embraced an open-ended approach of student-led activity for inquiry-based student problem-solving. After exploring the various flaws in ZigBee, Zwave, Bluetooth, and LoRa protocols, we granted remote access to a lab that contained approximately fifty different smart home IoT devices. These devices included cameras, doorbells, motion sensors, locks, environmental sensors, lights, and digital assistants. The instructor challenged the students to identify and research an individual device for potential security vulnerabilities based on their understanding of previous vulnerabilities. Students accessed remote terminals with the hardware described in Section 4.3, capable of monitoring devices. We observed increased student anxiety during this phase as students felt the pressure to solve an open-ended problem. We approached this problem carefully and partnered with students to help create experiments. As an example, we partnered with a student who wished to explore ZigBee key pre-provisioning by finding suitable equipment and software to eavesdrop on a ZigBee device pairing. Recognizing the constraints of remote learning, we took a more deliberate coaching approach during this lab than we would have during in-person learning.

4.2.6 Final Lab: Final Lightweight Protocol (FLiP). By the end of the five labs, students understood secure designs for mobile and

Listing 2: Simulating frames via a GnuRadio docker container helped overcome the challenge of remote students.

```
$ nc challenges.ctfd.io 30394 < replay-dos.bin
[+] Welcom3 to FLiP Simulator v1.01
---------- Simulator Results ----------
aabb > c0d3 [Seq: 1000]  (Len: 4) Echo Request: aGVsbG8K
aabb > c0d3 [Seq: 1000]  (Len: 4) Echo Request: aGVsbG8K
aabb > c0d3 [Seq: 1000]  (Len: 4) Echo Request: aGVsbG8K
aabb > c0d3 [Seq: 1000]  (Len: 4) Echo Request: aGVsbG8K
aabb > c0d3 [Seq: 1000]  (Len: 4) Echo Request: aGVsbG8K
>> Identified five frames with src: aabb and seqnum: 1000
>> Implementing FLiP privacy protection (FLiPPP).
>> flag{Things are only impossible until they are not.}
```

wireless protocols. The final lab aggregated the course concepts and offered an opportunity to demonstrate synthesizing the course knowledge. The instructor prepared a specification and implementation for a novel RF protocol using the GnuRadio software. The specification contained several design flaws, including pre-provisioned keying, susceptibility to known-plaintext accounts, vulnerability to denial of service attacks, and sensitive information disclosure. These vulnerabilities mirrored examples from the course topics, including ZigBee's pre-provisioned key problem, Microsoft's flawed RF keyboard XOR encryption, Bluetooth's address disclosure, and 802.11 unencrypted management frames. The instructor distributed the specification and a software library to receive and transmit frames the week before the lab. Students could make their attack toolkits, rehearse eavesdropping and processing frames, and thoroughly examine the specification for flawed security designs in advance of the lab. During the lab, students decrypted provided signal samples and synthesized samples to force attacks on a remote simulator. The simulator server offered remote connectivity by processing GnuRadio file sinks, running the protocol library inside a docker container. Upon a successful attack, the server announced a flag as depicted in Listing 2.

4.3 Course Infrastructure

Embracing hands-on labs with remote learning required significant course infrastructure. We capture the radio systems we used in Table 2. Due to the varying frameworks and software requirements to implement and attack different protocols, we required eight unique radio systems. We distributed low-cost RTL-SDR radios, RaTL Snake antennas, and Panda WiFi dongles to students to conduct remote experiments at their homes. Groups used the HackRF radios when they needed to synthesize and transmit a signal. The instructor used the BladeRF (a full-duplex transceiver) to construct a GSM base station in the classroom. Other radios served application-specific purposes with the Attify Mote, YardStick One, Ubertooth, and CrazyRadio radios flashed with firmware to attack ZigBee, Zwave, Bluetooth, and RF keyboard protocols, respectively. The instructor installed these application-specific radios on lab terminals available through a classroom VPN tunnel. We also made network-capable USRP N210s radios available on the VPN tunnel.

Table 2: Nine different radio systems were required to implement and attack different RF protocols.

Device	Purpose
RTL-SDR Dongle	Sub-1 GHz low-cost SDR receiver
Panda WiFi 802.11 Dongle	802.11 monitor-mode capable radio
HackRF SDR	Half-duplex SDR
BladeRF SDR	Full-duplex SDR
Attify Mote	KillerBee capable radio
Yardstick One	KillerZee capable radio
Ubertooth One	Bluetooth/BLE capable radio
CrazyRadio Dongle	Mousejack capable radio
USRP N210	Networked full-duplex SDR

These networked resources allowed remote students to interact with the lab as if they were physically present in the classroom.

5 LESSONS LEARNED

This section shares the successes and challenges we experienced teaching a hands-on mobile and wireless cybersecurity course.

5.1 Successes

Students Enjoy Attacking Themselves: A key insight of our course was that students enjoy the experience of attacking physical devices in their environments. We first observed this in the rapid radio reversing lab. Students shared Snapchat videos of their work, identifying their analysis of car key fobs, garage door openers, doorbells, power outlets, and remote-controlled cars. Although the assignment only called for the study of a single device, enthusiasm for the course material resulted in every team analyzing multiple devices. The same enthusiasm fueled the student discovery and reporting of four Common Vulnerability and Exposures (CVEs) during the IoT Firmware and RF Protocol Investigation Labs. Although preparing hands-on experiences presented challenges, our interactions with students showed they were sincerely motivated by and appreciated this approach.

Leaving on a High Note: Students gained an authentic experience reading through a protocol specification, identifying novel vulnerabilities, and developing methods and tools to attack these vulnerabilities in the final lab. This dual-edged approach worked as the final test to measure the student's ability to synthesize knowledge while also leaving the student with a sense of empowerment after the course. Both in discussions and exam submissions, student commented that they felt empowered to investigate vulnerabilities against the confidentiality and authentication of a wireless protocol. The exam included two static questions with instructor provided traffic and two dynamic questions that required the students to craft their traffic on a simulator. Students performed better on the dynamic questions by scoring a 99.06% average on these questions, which tested divergent approaches for students analyzing, evaluating and attacking the protocol. Students. In contrast, student grades averaged 84.43% on static problems that required them to analyze and apply an attack against the instructor-provided recordings. We hypothesize that this may be due to students enjoyed seeing the

impact of their attack in the simulator versus discovering a vulnerability in static file processing. As future iterations of the course will be conducted in-person, we look forward to examining how implementing the protocol on actual hardware will change student interest.

5.2 Challenges

Mixing Adversary Thinking and Cooperative Learning: One of the critical challenges of conducting cooperative learning is an instructor's ability to judge individual student involvement. The students in our class had varying degrees of prior knowledge, skills, and abilities. The aspect of remote learning further amplified the challenge of judging individual effort. Throughout the course, we tried different strategies to enforce effective group-work and peer learning. However, we observed that the success of each group's adversary thinking typically relied on a single individual's ability to embrace unconventional perspectives and reason. Twenty-five percent of our students possessed these traits, judging by their performance on the individual capture-the-flag competition. We rotated these students among different teams for each lab to facilitate peer-learning. To ensure student grades measured student outcomes, we conducted the final lab as an individual event. We reserve an analysis of adversary thinking in cooperative learning environments for future work.

You Can Only Solve Failures You've Rehearsed: Virtual and remote learning can minimize student social anxiety [16]. However, we discovered that encountering technical challenges during virtual learning can magnify both student and instructor anxiety. For example, we spent twenty class minutes debugging a troublesome issue with a GnuRadio block before discovering the problem in one particular instance. We only solved the case after finding the GnuRadio block was detecting a chipset on the student's RaspberryPi instead of the student-provided radio. Following this negative experience, we began *rehearsing failures*. Before each meeting, we would attempt to identify how students might break the hands-on lesson or lab. We rehearsed the failure and recorded outcomes for each type of mistake. We prepared a table of the symptoms of technical failures for each meeting to allow rapid troubleshooting and minimizing student anxiety through this approach. We observed that rapidly fixing student technical issues and explaining the underlying problem bolstered student confidence and reduced anxiety.

Standardizing the Environment: A semester-long hands-on examination of mobile and wireless protocols presents a challenge. Typically, an instructor can overcome these challenges by providing a standardized classroom lab for experiments. However, amid the global pandemic, our university chose to offer classes remotely. This challenge placed an additional burden on both students and instructors to ensure access to resources. Further complicating this problem is that several proof-of-concept tools often rely on deprecated libraries or software. For example, our environment required three different Python instances, ranging from 2.7 to 3.8, to support everything from the abandoned KillerZee framework to the current GnuRadio release. After trial and error, we discovered the best method for providing access to resources was to distribute a standard virtual machine complete to our students. This method allowed the instructor to control the environment and better anticipate and troubleshoot student problems, as discussed in the previous paragraph.

6 CONCLUSION

In this paper, we presented our undergraduate mobile and wireless cybersecurity course. Our course balances theory-based instruction and experiential learning in a cooperative environment. Our course consists of inter-woven lectures with labs alongside a semester-long capture-the-flag competition. Students use software-defined-radios and open-source software to perform contemporary attacks against radio-frequency RF-enabled hardware, IoT devices, and mobile and wireless protocols. The course culminated with students attacking a flawed RF protocol implemented on GnuRadio to allow students to demonstrate their knowledge synthesis. Due to the impact of the novel coronavirus, we conducted the class via a remote learning model. We have shared our experiences, hoping to offer insight for instructors who wish to embrace adversarial thinking in a mobile & wireless cybersecurity course while overcoming the challenge of remote students.

ACKNOWLEDGEMENTS

This material is based upon work supported in whole or in part with funding from the Office of Naval Research (ONR) contract #N00014-20-1-2798. 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] Daniele Antonioli, Nils Ole Tippenhauer, and Kasper B Rasmussen. 2019. The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX, Santa Clara, CA, 1047–1061.

[2] John Aycock, Andrew Groeneveldt, Hayden Kroepfl, and Tara Copplestone. 2018. Exercises for teaching reverse engineering. In *23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. ACM, Larnaca Cyprus, 188–193.

[3] Christopher W Badenhop, Scott R Graham, Benjamin W Ramsey, Barry E Mullins, and Logan O Mailloux. 2017. The Z-Wave routing protocol and its security implications. In *Computers & Security*, Vol. 68. Elsevier, 112–129.

[4] 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. In *IEEE Transactions on Emerging Topics in Computing*, Vol. 4.3. IEEE, 429–438.

[5] Miguel Bazdresch. 2016. Considerations for the design of a hands-on wireless communications graduate course based on software-defined radio.. In *2016 IEEE Frontiers in Education Conference (FIE), Frontiers in Education Conference (FIE), 2016 IEEE*. IEEE, 1 – 5.

[6] Ian Beer. 2020. An iOS zero-click radio proximity exploit odyssey. https://googleprojectzero.blogspot.com/

[7] Fabrice Bellard. 2005. QEMU, a fast and portable dynamic translator.. In *USENIX Annual Technical Conference, FREENIX Track*, Vol. 41. USENIX, Anaheim, CA, 46.

[8] Anastasios N Bikos and Nicolas Sklavos. 2012. LTE/SAE security issues on 4G wireless networks. In *IEEE Security & Privacy*, Vol. 11.2. IEEE, 55–62.

[9] Alex Biryukov, Adi Shamir, and David Wagner. 2000. Real Time Cryptanalysis of A5/1 on a PC. In *International Workshop on Fast Software Encryption*. Springer, Springer, Paris, France, 1–18.

[10] 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.

[11] Peter Chapman, Jonathan Burket, and David Brumley. 2014. PicoCTF: A game-based computer security competition for high school students. In *2014 USENIX*

Summit on Gaming, Games, and Gamification in Security Education (3GSE 14). USENIX, San Diego, CA.

[12] Daming D Chen, Maverick Woo, David Brumley, and Manuel Egele. 2016. Towards Automated Dynamic Analysis for Linux-based Embedded Firmware.. In *Network and Distributed System Security Symposium (NDSS)*, Vol. 16. The Internet Society, San Diego, CA, 1–16.

[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.

[14] Kevin Chung. 2017. Live Lesson: Lowering the Barriers to Capture The Flag Administration and Participation. In *2017 USENIX Workshop on Advances in Security Education (ASE 17)*. USENIX, Vancouver, BC, Canada.

[15] Brian P Crow, Indra Widjaja, Jeong Geun Kim, and Prescott T Sakai. 1997. IEEE 802.11 wireless local area networks. In *IEEE Communications magazine*, Vol. 35.9. IEEE, 116–126.

[16] Jelia R Domingo and Elizabeth Gates Bradley. 2018. Education student perceptions of virtual reality as a learning tool. In *Journal of Educational Technology Systems*, Vol. 46.3. SAGE Publications Sage CA: Los Angeles, CA, 329–342.

[17] Behrang Fouladi and Sahand Ghanoun. 2013. *Security evaluation of the Z-Wave wireless protocol.* Technical Report. Black Hat. 1–2 pages.

[18] Elaine M Grossman. 2006. *Millennium Challenge'02 lessons left unresolved: Three Years Later, A Retired General Awaits Joint Experiment Report.* Technical Report 21. Inside the Air Force. 5–7 pages.

[19] 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. In *IEEE Transactions on Education*, Vol. 60.3. IEEE, 205–211.

[20] Russell Handorf. 2014. DEFCON SDR WCTF. http://sdr.ninja/training-events/sdr-wctf/

[21] Craig Heffner. 2010. Binwalk firmware analysis tool, 2010. , 2017 pages.

[22] Maurice Hendrix, Ali Al-Sherbaz, and Bloom Victoria. 2016. Game based cyber security training: are serious games suitable for cyber security training?. In *International Journal of Serious Games*, Vol. 3.1. Serious Games Society.

[23] Darren Kitchen. 2017. Hak5 2120 – Shmoocon 2017: Sniffing IR Signals and More! https://www.hak5.org/episodes/hak5-2120-shmoocon-2017-sniffing-ir-signals-and-more

[24] Angela M Lonzetta, Peter Cope, Joseph Campbell, Bassam J Mohd, and Thaier Hayajneh. 2018. Security vulnerabilities in Bluetooth technology as used in IoT. In *Journal of Sensor and Actuator Networks*, Vol. 7.3. Multidisciplinary Digital Publishing Institute, 28.

[25] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux journal* 2014, 239 (2014), 2.

[26] MITRE. 2013. CVE-2014-0160. Available from MITRE, CVE-ID CVE-2014-0160.. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160

[27] MITRE. 2019. CVE-2019-9483. Available from MITRE, CVE-ID CVE-2019-9483.. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9483

[28] MITRE. 2020. CVE-2020-24490. Available from MITRE, CVE-ID CVE-2020-24490.. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-24490

[29] MITRE. 2020. CVE-2020-29439. Available from MITRE, CVE-ID CVE-2020-29439.. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-29439

[30] Stig Fr Mjølsnes and Ruxandra F Olimid. 2017. Experimental Assessment of Private Information Disclosure in LTE Mobile Networks.. In *International Conference on Security and Cryptography (SECRYPT)*. ICSC, Madrid, Spain, 507–512.

[31] Philipp Morgner, Stephan Mattejat, and Zinaida Benenson. 2016. All your bulbs are belong to us: Investigating the current state of security in connected lighting systems. In *arXiv preprint arXiv:1608.03732*. arXiv.

[32] Karsten Nohl. 2010. *Attacking phone privacy.* Technical Report. Black Hat USA. 1–6 pages.

[33] NSA. 2020. 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/

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

[35] Michael Ossmann. 2016. *Rapid radio reversing.* Technical Report. Great Scott Gadgets.

[36] Chris Paget. 2010. *Practical cellphone spying.* Technical Report. DEFCON 18 Security Conference.

[37] Johannes Pohl and Andreas Noack. 2018. Universal radio hacker: a suite for analyzing and attacking stateful wireless protocols. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. USENIX, Baltimore, MD.

[38] Moe Rahnema. 1993. Overview of the GSM system and protocol architecture. In *IEEE Communications magazine*, Vol. 31.4. IEEE, 92–100.

[39] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. 2020. Call Me Maybe: Eavesdropping Encrypted LTE Calls With ReVoLTE. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX, Online, 73–88.

[40] SAGE. 2011. *Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC specification.* Technical Report. ETSI.

[41] Cem Sahin, Danh Nguyen, James Chacko, and Kapil R Dandekar. 2015. Wireless cybersecurity education via a software defined radio laboratory. In *2015 IEEE Frontiers in Education Conference (FIE)*. IEEE, El Paso, TX, 1–8.

[42] Thorsten Schroeder and Max Moser. 2010. *Practical exploitation of modern wireless devices.* Technical Report. CanSecWest.

[43] 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.

[44] Dominic Spill and Andrea Bittau. 2007. BlueSniff: Eve Meets Alice and Bluetooth.. In *Workshop on Offensive Technologies (WOOT)*, Vol. 7. 1–10.

[45] 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.

[46] 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.

[47] 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.

[48] Mathy Vanhoef and Frank Piessens. 2014. Advanced Wi-Fi attacks using commodity hardware. In *30th Annual Computer Security Applications Conference*. ACSAC, New Orleans, LA, 256–265.

[49] Mathy Vanhoef and Frank Piessens. 2017. Key reinstallation attacks: Forcing nonce reuse in WPA2. In *2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Dallas, TX, 1313–1328.

[50] Mathy Vanhoef and Frank Piessens. 2018. Release the Kraken: new KRACKs in the 802.11 Standard. In *2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Toronto, Canada, 299–314.

[51] Mathy Vanhoef and Eyal Ronen. 2020. Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. In *2020 IEEE Symposium on Security and Privacy-S&P 2020)*. IEEE, Online.

[52] 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.

[53] Denise Wilson, Diane Jones, Fraser Bocell, Joy Crawford, Mee Joo Kim, Nanette Veilleux, Tamara Floyd-Smith, Rebecca Bates, and Melani Plett. 2015. Belonging and academic engagement among undergraduate STEM students: A multi-institutional study. In *Research in Higher Education*, Vol. 56.7. Springer, 750–776.

[54] Joshua Wright. 2009. *Killerbee: practical zigbee exploitation framework.* Technical Report. ToorCon.