HELO DarkSide: Breaking Free From Katas and Embracing the Adversarial Mindset in Cybersecurity Education

TJ OConnor
toconnor@fit.edu
Florida Institute of Technology
Melbourne, FL, USA

ABSTRACT

The pedagogy of cybersecurity education presents an exciting challenge. Although cyber-warfare has existed for nearly four decades, we fail to adequately model the chaos of offensive cyber attacks in the classroom. Instead, coursework focuses on studying choreographed cyber-attack patterns. In this paper, we present an undergraduate cybersecurity course design that balances theoretical learning with an emphasis on exploring offensive tactics, techniques, and procedures. Labs consist of writing payloads and channels to evade detection, cobbling together operating system internals to achieve attack functionality, and developing survivable post-exploitation tools. In the exams, students develop malware capable of avoiding static and dynamic analysis and identify the strategic and tactical flaws that lead to the discovery of highly successful attack campaigns. We believe that sharing this experience will prove valuable for instructors who wish to explore offensive cyber tactics in the classroom.

CCS CONCEPTS

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

KEYWORDS

cybersecurity education, malicious software, adversary mindset

ACM Reference Format:
TJ OConnor. 2022. HELO DarkSide: Breaking Free From Katas and Embracing the Adversarial Mindset in Cybersecurity Education. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3478431.3499404

1 INTRODUCTION

In December 2020, the US Cybersecurity and Infrastructure Security Agency (CISA) released the details of a crippling state-sponsored cyber attack. The attack, perpetrated by Russian hackers, compromised the supply chain of the widely-used SolarWinds network monitoring software [8]. The attackers updated the SolarWinds software to include a time-delayed backdoor that established a stealthy command and control channel [1]. After activating the backdoor, the attackers hunted the networks of a subset of SolarWinds' 300,000 government, military and private sector customers. After a series of lateral moves, the attackers accessed and exfiltrated sensitive materials from the victims. In an era of pervasive attacks, the Russian attackers distinguished themselves for the attack's sophistication and broad reach. Through a deliberate planning and development process, the attackers largely evaded and outsmarted security vendors and intelligence agencies until the end-stage of the campaign [26].

We hypothesize that current cybersecurity education objectives and paradigms contribute to an unprepared cybersecurity workforce. Narrowly focused coursework on *ethical hacking* fails to introduce our students to the chaos of the adversarial mindset and real-world attack campaigns. In a similar problem, the traditional Karate martial arts discipline has been criticized for teaching repeated movement patterns. These choreographed movements, or katas, offer the opportunity to perfect the form and technique through repeated deliberate practice. However, they often install a closed-minded approach where a karateka struggles against outside techniques such as in mixed-martial-arts (MMA) fights [4]. For high-level karatekas to evolve in MMA, they found it necessary to modify techniques and engage in *yakusoku kumite* (sparring) [12]. We hypothesize that current cybersecurity education parallels Karate, teaching a closed taxonomy of attack techniques. This approach results in producing a workforce incapable of reacting to outside-the-system techniques such as the SolarWinds attack.

This paper shares a detailed description of the design, labs, and exams for an undergraduate cybersecurity course on offensive cyber tactics. We offer our approach, developing students by combining lessons on the MITRE Att&ck Framework tactics with exploratory lab opportunities. Further, we examine the ethical considerations of implementing such as curriculum. This paper makes the following contributions:

(1) In light of recent pervasive attack campaigns, we examine the responsibility and the methods for cybersecurity education to examine offensive cyber tactics.
(2) We share our experiences, lessons learned, and materials, emphasizing the MITRE Att&ck framework to overcome the challenges of decaying content.

Organization: Section 2 investigates previous work. In Section 3, we provide an overview of our course and the design approach. Section 4 examines the course labs. Section 5 offers insight and examines future challenges. Section 6 summarizes our conclusions.

SIGCSE 2022, March 3–5, 2022, Providence, RI, USA

ACM ISBN 978-1-4503-9070-5/22/03…$15.00
https://doi.org/10.1145/3478431.3499404

2 PRIOR WORK

Our paper advocates for the responsibility of educators to prepare students to enter the cyber workforce with the capability to model and implement offensive tactics. As such, we first review some federal directives in this area. The Cybersecurity and Infrastructure Security Agency (CISA) recently released guidance for professional development in the Cyber Career Pathways Tool [18]. This limited framework describes the knowledge, skills, and abilities for 52 work roles for cyber professionals. However, it is broadly focused and lacks the fine-grained detail for a workforce under pervasive attack. In contrast, the National Security Agency Center of Academic Excellence Cyber Operations (NSA CAE CO) knowledge units offer a deeply technical, inter-disciplinary set of skills [20]. The NSA CAE CO curriculum is grounded firmly in computer science, computer engineering, and electrical engineering. However, the NSA CAE CO suffers from limited adoption, as less than 30 institutions in the US have achieved the NSA CAE CO designation [20]. We offer our course design as a curriculum designed to meet a subset of the NSA CAE CO outcomes and objectives. The National Institute of Standards and Technology (NIST) leads a broader National Initiative for Cyber Education (NICE) partnership between academia, government, and the private sector to focus on cybersecurity education. While NICE has produced framework mappings to the NSA CAE Cyber Defense (NSA CAE CD) knowledge units, they do not offer a map to the deeply technical Cyber Operations (NSA CAE CO) knowledge units [19]. However, we acknowledge the challenges of maintaining finer details of these knowledge units as tactics, techniques, and procedures decay over time.

Further, we recognize that our work is not the first to advocate for introducing offensive techniques in the classroom [3, 15, 23, 30]. Trabelsi and Saleous argued that teaching keylogging and network monitoring proved essential to understanding the threat [29]. Bratus identified industry and academic trends that conflict with the offensive mindset [3]. He argued academic instructors oversimplify attack patterns without exploring alternatives or understanding the finer details. Manson et al. explored the concept of integrating cybersecurity competitions, developing the National Cyber League (NCL) to promote hands-on learning [15]. However, Shoemaker et al. have critiqued the NCL, arguing that NCL challenges fail to integrate pedagogical design. Instead of exploring the conceptual understanding of a network scan, students focus on tool usage [30]. In contrast, our work argues for the path exploration advocated by Bratus by challenging students to develop their own tools and attack patterns. We balance this approach by leveraging the MITRE Att&ck Framework [28] to avoid challenges with structure and decaying tactics.

3 COURSE OVERVIEW AND DESIGN

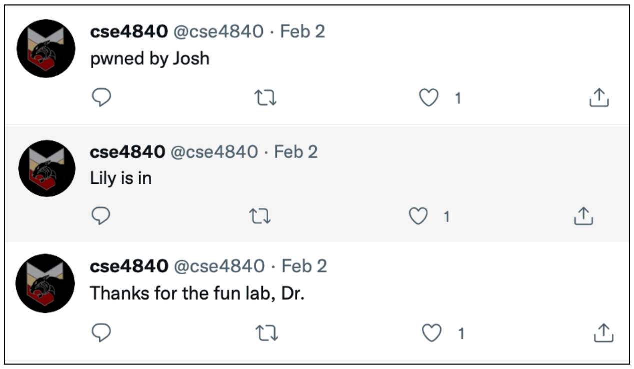
This section describes the course model for our *cyber offense* cybersecurity course. The course meets twice weekly for 75 minutes sessions over 15 weeks. We inter-weave weekly lessons on tactics with exploratory labs. The course culminates our six-course *cyber operations* concentration inside of an ABET-accredited Computer Science Bachelors program. The previous five courses present topics on cyber operations fundamentals, cyber defense, wireless security [23], binary reverse engineering [27], and vulnerability research. Prerequisites consist of traditional computer science courses, including assembly programming, data structures, object-oriented design, and operating systems. To support the following objectives, we anchor our course with the MITRE Att&ck Framework.

1. Students will be able to enumerate the phases of a cyber attack taxonomy, describe what each phase entails, and assess the impact of a cyber attack in the context.
2. Students will be able to replicate and model aspects of contemporary attacks by developing offensive techniques and procedures for evasion, control, and persistence.
3. Students will be able to discuss the trade-offs for tactical and technical planning considerations in a cyber attack.

3.1 Leveraging MITRE Att&ck Framework

We leverage the MITRE Att&ck framework to create the structure for studying the offense while avoiding the challenge of decaying content. This structure is necessary to remove the undefendable nature that often characterizes cyber attacks. Industry and academia perform a disservice by characterizing cyber attacks as the actions of invisible nation-state actors and criminal syndicates. After the recent ransomware attacks against the Colonial Pipeline and JBS Meat-Packing Plant, the public outcry echoed this sentiment. Both industry and government lamented the difficulty of defending against invisible adversaries. However, both the DarkSide Group attack sequences and target selection were predictable [1, 21]. DarkSide's well-documented approach relies on specific tools, including *Cobalt Strike* for establishing a foothold, *Mimikatz* for escalating, *BloodHound* for pivoting laterally, and *WinSCP* for data exfiltration [21]. Further, DarkSide targeted Colonial only after recovering virtual private network (VPN) credentials from the dark web in a well-known batch of exposed credentials. The compromised credentials and lack of multi-factor authentication proved a valuable invitation for the DarkSide group to attack Colonial. However, the steps, including reconnaissance, initial access, persistence, lateral movement, and data exfiltration, all followed typical DarkSide tactics.

Modeling these tactics proves helpful to predicting future attacks. Early work by Hutchins et al. proposed the *cyber kill chain* model to understand the goals and sophistication of computer network intruders [11]. Their work developed a model for identifying patterns that allows intelligence gathering to link individual intrusions into broader campaigns. The MITRE Att&ck framework expanded on this by systemically categorizing adversarial behavior into specific tactics, techniques, and procedures (TTPs) [28]. The framework consists of thirteen coarse-grained tactics iterating from reconnaissance to understanding the attack's impact. MITRE also continually updates an online matrix, documenting the current techniques and procedures spotted in the wild by adversaries. The matrix greatly simplifies the challenge of maintaining heavily applied course materials necessary for studying cyberwarfare [2]. We leverage each of the tactics categories as focus efforts for the weekly lessons, exploring the techniques and procedures involved in each tactics category.



cse4840 @cse4840 · Feb 2
pwned by Josh
1

cse4840 @cse4840 · Feb 2
Lily is in
1

cse4840 @cse4840 · Feb 2
Thanks for the fun lab, Dr.
1

Figure 1: We observed students' enthusiasm and excitement breaking out of the restricted environment, tweeting their results in the living of the land lab.

4 COURSE IMPLEMENTATION

This section describes the course implementation, including the course labs and exams. We follow a *lectures with labs model*, where each theory-based lesson corresponds with a practical lab. We explore the rationale behind our labs and assessments, leveraging recent attacks to justify our design.

4.1 Course Labs

4.1.1 Remote Access Toolkit (RAT) Lab. In the first lab, we challenged students to install and present a demonstration of a remote access toolkit. Malicious actors commonly employ remote access toolkits (RATs) to execute their malware, move laterally in target environments, and exfiltrate data. We began our labs with this unorthodox approach to break free from the misunderstanding that presents cyber attacks as complex and chaotic actions. Reviewing the tactics of the recent DarkSide ransomware group attack on the Colonial Pipeline presents an interesting observation. The DarkSide group, similar to most criminal syndicates, heavily leveraged the *Cobalt Strike* toolkit to execute, pivot through the network, and exfiltrate sensitive files [21]. While Cobalt Strike is a sophisticated cyber weapon, it proves relatively easy to detect when network defenders understand its capability [9]. Similarly, this lab challenged students to discuss methods for defense. Students chose popular open-source tools, including Kodiac, PoshC2, Pupy, QuasarRAT, and Powershell Empire [16]. They examined how the tools abused protocols like DNS or HTTP to hide their control channels. Further, they discovered how the tools employed file-less attacks, relying on Powershell scripting to launch attack functionality. An interesting observation is that most tools required students to disable protection mechanisms, including Windows Defender, Powershell Script Execution Policies, and Anti-Malware-Scan-Interface (AMSI). This lab provided a wealth of context for the students, strengthening their theoretical study with a practical examination of actual attack tools.

4.1.2 Living Off the Land Lab. The following lab explored living-off-the-land (LOTL) execution tactics. Recently, LOTL tactics have benefited cyber-espionage attack campaigns [24]. Rather than introducing malware or post-exploitation tools to the target environment, the attackers cobble together existing operating system binaries to perform attack functions. Attackers abuse the binaries' unintended functionality to download, upload, read, write files, load, and execute binaries to break out of restricted environments. This creative tactic succeeds in strictly monitored targets because the attacker uses the victims' tools against themselves. For the living off the land lab, we prepared a docker container with a limited set of binaries that restricted the functionality of the students. Further, we placed an encoded file containing a secret (a link to a Python script) in the environment. After pushing the container to a registry and enabling remote access, we challenged the students to escape the restricted environment by 1) reading the restricted file, 2) decoding the contents of the file, 3) downloading a separate Python script, and 4) executing the script on the target. We removed any operating system tools that directly performed these tasks. For example, we uninstalled the Python interpreter binary from the system. Despite these challenges, all the students succeeded in escaping the restricted challenge. They discovered several unique and creative solutions, such as using the Python interpreter bundled with the gdb debugger. As a playful conclusion of the exercise, the downloaded script allowed the students to publish a tweet to the course Twitter account. As depicted in Figure 1, the student messages relayed excited messages, praising the lab and sharing their excitement of breaking out of the restricted environment.

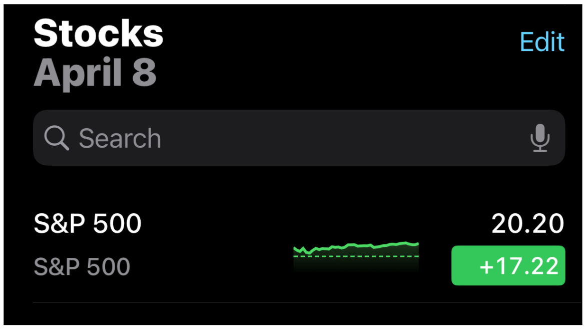
Listing 1: Students learned about the asymmetric nature of the offense. As an example this student evaded antivirus simply by encoding every other byte in her shellcode.

```
lea rsi, [shellcode]                    # load shellcode
mov rcx, 0x00
mov bl, 0x00
loop:                                   # iterate through bytes
    mov al, byte [rsi+rcx]
    test rcx, 1
    jz even
    even:
        add al, bl                      # encode even bytes
        add bl, 2
        jmp short continue
    continue:
        mov byte [rsi+rcx], al
        inc rcx
        cmp rcx, length
        jne loop
```

4.1.3 Evasive Shellcode Lab. Our next lab explored defensive evasion by encoding shellcode to evade detection. A payload (i.e., shellcode) is a critical component of any remote code execution (RCE) attack that targets a service. By its very nature, the payload performs the malicious intentions of the attacker. Payloads are often referred to as shellcode as they provide a remote shell or session to the victim. To detect shellcode, security operations centers rely on network monitoring and endpoint detection systems with well-developed signatures. In response, attackers modify their shellcode to evade detection. In the shellcode detection lab, we challenged students to bypass these systems by developing a unique shellcode encoding method. We provided the students with well-known and detectable shellcode. We demonstrated that antivirus engines could also detect commonly used evasive approaches such as modifying every byte with an XOR operation. Students attempted several methods to overcome detection, experimenting with different techniques for manipulating and encoding their shellcode bytes. Students discovered that hypothesizing how defensive mechanisms worked proved most helpful in overcoming detection. One student hypothesized that the detection engines most likely identified the XOR-encoding approach by matching the instruction sequence that performed this loop and encoding. She hypothesized that modifying this sequence could avoid detection. Her experiment is depicted in Listing 1. Scanning the constructed shellcode, she proved her hypothesis that manipulating only the even bytes evades detection. This result emphasized a critical course theme that well-intentioned defenses often fail to capture the creative adversarial approach that scrutinizes border cases to exploit.

4.1.4 Collection Lab. Our next lab explored collection tactics that occur after exploitation. During the collection phase of an attack, adversaries gather information and sources of information based on their campaign objectives and goals. Attackers target rich data sources, including browser and email artifacts, to collect and exfiltrate off the system. In red teaming, network penetration testers refer to this phase as the *post-exploitation* phase since it occurs after the successful compromise of a system. We challenged students to write a *post-exploit* module for the popular open-source *Metasploit* exploitation framework. A well-developed framework, *Metasploit*, has an existing Ruby-based application programming interface that allows for contributors to develop new modules [13]. Our students wrote modules to gather evidence from email and browser clients, exfiltrate recently downloaded files, and application logs. One student developed a very creative module to parse through developers' version control logs, identifying *.gitignore* files that contained sensitive information. We offered extra credit for students who took the additional effort to push their modules to the Metasploit official repository. We observed students' excitement grew as they waited for the Metasploit developers to review and accept their contributions to the framework. We identified that this excitement demands further study about the pedagogical benefit of contributing to open-source penetration testing tools in the classroom.

4.1.5 Man-in-the-middle Lab. In the next lab, we challenged students to explore weaknesses in cryptographic schemes that allow man-in-the-middle attacks. Despite the widespread popularity, smartphone applications commonly suffer poor cryptographic policy enforcement. In particular, several applications fail to pin encryption certificates. By bundling certificates for trusted servers with the application, certificate pinning prevents man-in-the-middle attacks (MiTM). However, previous works have demonstrated vendors commonly ignore certificate pinning in actual practice [10, 22]. In the man-in-the-middle attack lab, we challenged students to explore the implications of these attacks against smartphone applications by creating a *mitmproxy* script to intercept and manipulate applications [7]. This lab benefits from our creation of an IoT Security and Privacy Lab [5]. Several students focused their efforts on companion applications for IoT devices, spoofing a false state of the devices. They demonstrated how they could replace security camera images or falsely present a door as locked. Other students focused on applications that relayed critical information, such as the Apple Stocks application depicted in Figure 2. Despite this lab occurring near the end of the semester of attacks, we observed a great deal of student excitement and genuine surprise. We realized that this was the first lab that demonstrated attacks on real environments and devices. Attacking real-world devices presents a substantial risk in the classroom. However, the unique nature of this attack vector only enabled students to manipulate end-user content and prohibited targeting any external devices or remote servers. Further, we directed students to share their findings with the affected vendors in vulnerability reports. Due to the widespread and unexpected student surprise, this lab generated, we identify that attacking real-world devices in the classroom demands further study to balance student excitement with potential risks.


Stocks
April 8

S&P 500 20.20
S&P 500 +17.22

Figure 2: We observed students' surprise when they attacked actual smartphone applications in the man-in-the-middle lab. In this example, the student spoofed a crash of the S&P 500 on the default Apple Stocks application.

4.2 Exams

4.2.1 Antivirus Evasion Midterm Exam. For the midterm exam, we challenged students to develop a keylogger that relayed keystrokes to a remote attacker. While previous lessons examined hooking operating system internals, we purposely avoided directing students on how to complete the task. Instead, we specified constraints that the keylogger must evade detection, remain persistent, and bypass traditional operating system access controls. Students developed the assignment in various programming languages, including Rust, Nim, C, Python, and C#. Students discovered that antivirus engines struggled with signature-based detection methods for less popular languages such as Nim [25]. The students also observed several false positives trying to construct binaries from Python code. Further, they observed that different compilers produced different results. As depicted in Table 1, one student observed that using the Nuitka Compiler versus the Py2Exe compiler produced drastically different results. This finding reinforced a course theme that the

Table 1: An example of a student's results that tested the accuracy of antivirus engines and identified a method for reducing true positive detection of malware simply by changing the compiler.

AntiVirus Engine	Hello World	Py2Exe Compiled Keylogger	Nuitka Compiled Keylogger
Antiy-AVL	✗	✗	✓
Cynet	✗	✗	✓
SecureAgex APEX	✗	✗	✗
Jiangmin	✗	✗	✓
Zillya	✗	✗	✓
ESET-MOD32	✓	✗	✓
Gridinsoft	✓	✗	✓
Kaspersky	✓	✗	✓
Microsoft	✓	✗	✓
ZoneAlarm	✓	✗	✓
Yandex	✗	✓	✓

✗- identified as malicious
✓- identified as benign

offense succeeds by exploring border cases. In contrast, students experimented with well-known evasion frameworks such as Veil and The Backdoor Factory only to find that they struggled to evade holistic platforms such as Virus-Total or Hybrid-Analysis.

A particularly ambitious student learned a valuable lesson, bypassing static analysis on Virus-Total a week before the due date, only to discover that a vendor saved his sample for later dynamic analysis in a sandbox. As a result, his invisible malware suddenly became detectable by all antivirus vendors on exam day. Students who waited to scan until the last minute or simply avoiding scanning their malware did not suffer the same fate. Students experimented with avoiding dynamic analysis by disabling the keylogger functionality when they detected a sandbox environment. Borrowing techniques from APT37, one student evaded sandbox detection by sleeping when it noticed the usually limited memory and disk size of a sandbox. Instead of calling the sleep() function (which could be patched in a sandbox environment), the student wrote an inefficient algorithm to sort and unsort a list of numbers into perpetuity. The creative approaches reflect the chaotic creativity of offense. We never explained in a lesson to patch the sleep() function this way. Instead, the student recognized this critical detail and proposed a divergent solution. In our observations of students, we realized that the exam forced students to synthesize the course materials since simply repeating patterns from previous attacks would be detected.

4.2.2 Campaign Analysis Final Exam. For the final exam, we tasked the students to present a campaign analysis study. Similar to a history course examining a military campaign, the exam forced the students to illustrate strategic and tactical decision-making. While we asked the students to explore the tactics, techniques, procedures (TTPs), and motivations, we also posed an interesting question. How was the attack discovered? The exam questioned the students to explain the deliberate planning and tactics that could have overcome this detection. We challenged the students to identify the campaign mistakes that lead to attack detection. We motivated the exam with a discussion about the limitations of previous attacks.

computer science
valuable information
cyber classes
class size
bring
cares
demo
nice
week
conveys information
enthusiasm
semester
opportunities
feel
joy
amount
successful
effective
instructor
cyber offense
industry experts
industry
experience
learned
fun
makes learning enjoyable
larger classroom
people
talk
provided
vast amount
grad school
field
classes
forever grateful
great time
rat
learning
lab
helping
knowledge
class
students
provide
person learning
engaging
teaching
real malware
multiple guest speakers
properly instructed

Figure 3: Student surveys praised the real-world examples, contemporary attacks, and engaging experience of the course.

Our discussion examined the Stuxnet malware campaign that compromised the Iranian programmable logic controllers and ruined one-fifth of Iran's nuclear centrifuges [14] While it presents an interesting case study for its sophistication and motivations, the attackers made mistakes. At the time, reports boasted that the malware relied on four Windows zero-day vulnerabilities. However, when Belarusian antivirus expert Sergey Ulasen discovered Stuxnet in 2010, the malware still leveraged an older exploit that compromised the well-known MS08-067 vulnerability [17]. This mistake, arguable, lead to its detection and eradication. We asked the students to find other campaigns and historical technical errors that compromised the attacker's campaign plans. Students presented findings on criminal syndicates such as ransomware groups, nation-state actors with fine-grained tactical objectives, and intelligence organizations meant to persist. As the students completed their research, they gained a valuable understanding of the difficulty of managing the chaos and disorder of the offensive tactics to achieve campaign goals. While we directed students to synthesize their ideas into a twenty-minute presentation, most students took nearly double the allotted time as they hypothesized about the failures with supporting evidence.

5 LESSONS LEARNED

In this section, we explore our preliminary findings from the course. Following the course, We reviewed the end-of-course surveys. Figure 3 illustrates a word cloud of these results. Students praised the unorthodox experience. Their surveys reinforced our observations. Students described enjoying the real-world examples, contemporary attacks, and engaging experience. Despite this initial feedback, we explore the challenges we faced and continually observe the ethical considerations of the course.

5.1 Challenges

We acknowledge that decaying content proves the most significant hurdle to any cyber offense course [2]. We took great care to prepare lessons with recent attacks to overcome this hurdle. However, such an approach inevitably fails as defense evolves. Our first failure came within in the first few weeks of the course when showing a method for bypassing Microsoft's Anti Malware Scan Interface (AMSI). AMSI scans scripts for malicious code execution. When preparing a lesson on execution tactics, we included a drill-down into techniques for bypassing AMSI's restrictions. However, as the

AMSI defense tool evolved in the early weeks of the course, our bypass mechanism no longer evaded AMSI. Instead of removing this component of the lesson, we included the failed evasive technique. In presenting the material to the students, we note the original bypass mechanism and then identified the defensive evolution. Such discussions prove valuable to examining the continually rotating balance of power between offense and defensive techniques.

5.2 Ethical Considerations

Significant legal and ethical concerns accompany any course that incorporates the development or experimentation of malicious tools [29]. It is important to note that our course occurs as the sixth course in a six-course (two-year) sequence on cyber operations. In the first course, we present the students with an understanding of the Computer and Fraud Abuse Act (CFAA), Electronic Communications Privacy Act (ECPA), the Digital Millennium Copyright Act (DCMA), and our university's acceptable use policy. Students re-sign an ethics contract every semester that articulates the students' responsibilities and the unambiguous repercussions of violating policies and laws, as recommended in [6]. As a smaller university, we balance this with frequent appeal-to-reason discussions with the students about the risk student actions pose to the overall survivability of the program. However, we acknowledge the substantial risk that a former or current student may abuse skills, knowledge, or abilities gained in the course.

We balance this risk with the consideration of what happens if we fail to teach offensive tactics. Criminal syndicates and malicious nation-state actors do not share the same impasse when learning their craft. Rather, they advance whole-heartedly into the discipline of offensive computing. How can we expect to prepare the current cybersecurity workforce without digging deep into the offense? We argue that the classroom must expose students to these modern tactics. To enter the workforce and combat our current adversaries, students must write malware, command and control channels, evade antivirus, and manipulate network traffic. We cannot expect the current struggling workforce to educate students. Despite the risk, we are compelled to prepare students to fight the adversary by educating them on the same tactics, techniques, and procedures of criminal syndicates and nation-state adversaries.

5.3 Future Offerings

Student observations and evolving adversary tactics inform future offerings of our course. While students praised the course, we observed the greatest student enthusiasm and excitement focused in two areas. First, students enjoyed contributing to open source networking security platforms, such as in the *post-exploit* lab. As we prepare future iterations of this course, we will examine opportunities for students to contribute to open-source security tools. Further, we would like to explore how open-source contribution helps to mitigate imposter syndrome and reinforce students' self-efficacy. Second, students really enjoyed attacking real-world devices in the IoT *man-in-the-middle* lab. In future course offerings, we will look for further opportunities to leverage our IoT Security and Privacy Lab for practical attacks against real-world devices. Finally, evolving adversary tactics demand a routine investigation and course materials update. As adversaries place a greater weight on unique exploitation vectors, we must pursue an understanding of them in our materials. As an anecdote, we intend to pursue a greater understanding of *zero-click* mobile device attacks in the next course iteration. This addition is informed the MITRE Att&ck Framework Mobile Matrix and the worldwide penetration of mobile phone targets by the NSO Group.

6 CONCLUSION

In this paper, we presented our undergraduate course on cyber offense tactics. Our course consists of inter-woven lectures with labs. We present a course design that balances theoretical learning and exploratory labs. Students profit from experiential learning, writing payloads and channels to evade detection, cobbling together operating system internals to achieve attack functionality, developing survival post-exploitation tools, and manipulating network traffic. We anchor the course in the MITRE Att&ck framework, drawing historical context from real-world attack campaigns. We have shared our experiences, hoping to offer insight for instructors who wish to explore offensive cyber tactics in the classroom.

ACKNOWLEDGEMENTS

This material is based upon work supported in whole or in part with funding from the Office of Naval Research (ONR) award #AWD-000503. 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. Additionally, the author appreciates the students, including Sam Hartle, Liana Villafuerte, Robert Whitman, who shared samples of their work for this publication.

REFERENCES

[1] Andrew Archer, Doug Bienstock, Chris DiGiamo, and Glenn Edwards. 2020. Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor. https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html

[2] 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 *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. ACM, Virtual Event, 1205–1211.

[3] Sergey Bratus. 2007. What Hackers Learn that the Rest of Us Dont́. In *IEEE Security & Privacy Magazine*, Vol. 54. IEEE.

[4] George J Buse. 2006. No holds barred sport fighting: a 10 year review of mixed martial arts competition. In *British journal of sports medicine*, Vol. 40. British Association of Sport and Exercise Medicine, 169–172.

[5] Daniel Campos and TJ OConnor. 2021. Towards Labeling On-Demand IoT Traffic. In *Cyber Security Experimentation and Test (CSET)*. USENIX, Virtual Event.

[6] Thomas Cook, Gregory Conti, and David Raymond. 2012. When good Ninjas turn bad: Preventing your students from becoming the threat. In *Proceedings of the 16th Colloquium for Information System Security Education*. CISSE, Lake Buena Vista, FL, 61–67.

[7] Aldo Cortesi, Maximilian Hils, Thomas Kriechbaumer, and contributors. 2010–. mitmproxy: A free and open source interactive HTTPS proxy. https://mitmproxy.org/ [Version 7.0].

[8] Cybersecurity and Infrastructure Security Agency (CISA). 2020. CISA Updates Alert and Releases Supplemental Guidance on Emergency Directive for SolarWinds Orion Compromise. https://us-cert.cisa.gov/ncas/current-activity/2020/12/19/cisa-updates-alert-and-releases-supplemental-guidance-emergency

[9] Cybersecurity and Infrastructure Security Agency (CISA). 2020. DarkSide Ransomware: Best Practices for Preventing Business Disruption from Ransomware Attacks. https://us-cert.cisa.gov/ncas/alerts/aa21-131a

[10] Hossein Fereidooni, Tommaso Frassetto, Markus Miettinen, Ahmad-Reza Sadeghi, and Mauro Conti. 2017. Fitness trackers: fit for health but unfit for security and privacy. In *International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, Philadelphia, PA, 19–24.

[11] Eric M Hutchins, Michael J Cloppert, Rohan M Amin, et al. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. In *Leading Issues in Information Warfare & Security Research*, Vol. 1. Academic Publishing International Limited, 80.

[12] Velimir Jeknić, Goran Kasum, and Miloš Stojković. 2017. Karate in MMA Analysis of Lyoto Macida's Career and Fighting Style. In *International Journal of Physical Education, Fitness and Sports*. IJPEFS, Tamil Nadu, India., 12–18.

[13] David Kennedy, Jim O'gorman, Devon Kearns, and Mati Aharoni. 2011. Metasploit: the penetration tester's guide.

[14] Ralph Langner. 2011. Stuxnet: Dissecting a cyberwarfare weapon. In *IEEE Security & Privacy*, Vol. 9. IEEE, 49–51.

[15] Daniel Manson and Anna Carlin. 2011. A league of our own: the future of cyber defense competitions. In *Communications of the IIMA*, Vol. 11. New Orleans, LA, 1.

[16] MITRE. 2021. MITRE Att&ck Sofware. https://attack.mitre.org/software/

[17] Teague Newman, Tiffany Rad, LLC ELCnetworks, John Strauchs, and LLC Strauchs. 2011. SCADA & PLC vulnerabilities in correctional facilities.

[18] NICCS. 2021. Cyber Career Pathways Tool. https://niccs.cisa.gov/workforce-development/cyber-career-pathways

[19] NICE. 2021. NICE Mapping Tool. https://niccs.cisa.gov/workforce-development/mapping-tool

[20] 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/

[21] Jordan Nuce, Jeremy Kennelly, Kimberly Goody, and Andrew Moore. 2021. Shining a Light on DARKSIDE Ransomware Operations. https://www.fireeye.com/blog/threat-research/2021/05/shining-a-light-on-darkside-ransomware-operations.html

[22] 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.

[23] TJ OConnor and Christopher Stricklan. 2021. Teaching a Hands-On Mobile and Wireless Cybersecurity Course. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*. ACM, Virtual Event, 296–302.

[24] Benjamin S Rivera and Rhena U Inocencio. 2015. Doing more with less: a study of fileless infection attacks.

[25] Marcello Salvati. 2021. OffensiveNim. https://github.com/byt3bl33d3r/OffensiveNim

[26] Congressional Research Service. 2021. SolarWinds Attack—No Easy Fix. https://crsreports.congress.gov/product/pdf/IN/IN11559l

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

[28] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. 2018. *MITRE att&ck: Design and philosophy*. Technical Report. MITRE.

[29] Zouheir Trabelsi and Heba Saleous. 2018. Teaching keylogging and network eavesdropping attacks: Student threat and school liability concerns. In *Global Engineering Education Conference (EDUCON)*. IEEE, Istanbul, Turkey, 437–444.

[30] Yien Wang and Jianhua Yang. 2017. Ethical hacking and network defense: choose your best network vulnerability scanning tool. In *31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, Taipei, Taiwan, 110–113.