A STUDY ON ETHICAL HACKING IN CYBERSECURITY EDUCATION
WITHIN THE UNITED STATES

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science

by
Jordan Chew
March 2024
COMMITTEE MEMBERSHIP

TITLE: A Study on Ethical Hacking in Cybersecurity Education within the United States

AUTHOR: Jordan Chew

DATE SUBMITTED: March 2024

COMMITTEE CHAIR: Dongfeng Fang, Ph.D.
Assistant Professor of Computer Science
and Computer Engineering

COMMITTEE MEMBER: Ayaan M. Kazerouni, Ph.D.
Assistant Professor of Computer Science
and Software Engineering

COMMITTEE MEMBER: Zachary N.J. Peterson, Ph.D.
Professor of Computer Science
ABSTRACT
A Study on Ethical Hacking in Cybersecurity Education within the United States
Jordan Chew

As the field of computer security continues to grow, it becomes increasingly important to educate the next generation of security professionals. However, much of the current education landscape primarily focuses on teaching defensive skills. Teaching offensive security, otherwise known as ethical hacking, is an important component in the education of all students who hope to contribute to the field of cybersecurity. Doing so requires a careful consideration of what ethical, legal, and practical issues arise from teaching students skills that can be used to cause harm. In this thesis, we first examine the current state of cybersecurity education in the United States through a holistic view of funding, certifications, and course offerings. We then offer a framework to navigate the ethical and legal issues of teaching offensive security, as well as serve as a technical reference of useful tools for configuring and conducting a course in ethical hacking. Together, these contributions can be a baseline for educators looking to create courses on ethical hacking topics.
I want to thank my parents, my sister, and the rest of my family for being the biggest part of my support system even when life was not going to plan. Thank you for being there to help me in the lows and celebrate with me in the highs.

I also want to acknowledge my grandfathers, who both passed away while I was writing this thesis. While they did not really comprehend the work I was doing, they both played instrumental roles in shaping me into the person I am today.

A special thank-you goes out to the past, present, and future members and officers of the Cal Poly Security Education Club (CPSEC). The club has been a big reason as to why I became interested in cybersecurity in the first place, and continued to be a place for me to learn and discuss cybersecurity throughout my time at college.

Additionally, I really appreciate my numerous friends and peers, both those in the CSC department and not, for motivating me, keeping me accountable, and helping me to enjoy my time in college.

I would like to thank Dr. Ayaan Kazerouni and Dr. Zachary N.J. Peterson for helping further my interest in the academic research side of computing education and computing security, as well as serving as committee members in my defense.

Finally, I want to thank my advisor, Dr. Phoenix Fang, for being an incredible inspiration and peer. I would have had a much more difficult time navigating my time in the Master’s program without your guidance and compassion.
TABLE OF CONTENTS

| LIST OF TABLES | x |
| LIST OF FIGURES | xi |
| LIST OF ACRONYMS | xii |

CHAPTER

1 Introduction ......................................................... 1

1.1 Problem ............................................................ 1

1.2 Contribution ....................................................... 3

1.3 Ethical Statement .................................................. 3

2 Background and Motivation ........................................... 5

2.1 Institutions for Cybersecurity Education ......................... 6

2.1.1 National Security Agency (NSA) ............................. 6

2.1.2 National Centers of Academic Excellence in Cybersecurity (NCAE-C) ................................................. 6

2.1.3 National Science Foundation (NSF) .......................... 7

2.1.4 DeterLab ......................................................... 7

2.2 Terms ......................................................................... 8

2.2.1 Capture the Flag .................................................. 8

2.2.2 White/Black Hat .................................................. 8

2.3 Related Work ........................................................... 8

2.3.1 Tool Comparisons ............................................... 9

2.3.2 Course Case Studies ............................................. 9

2.3.3 Developing Lab Environments ................................. 10
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.4</td>
<td>Ethical Concerns</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Motivation</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Methodology</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>University Education</td>
<td>12</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Aggregate Data</td>
<td>13</td>
</tr>
<tr>
<td>3.1.1.1</td>
<td>NSF Funding</td>
<td>13</td>
</tr>
<tr>
<td>3.1.1.2</td>
<td>CAE</td>
<td>14</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Survey of University Courses</td>
<td>16</td>
</tr>
<tr>
<td>3.2</td>
<td>Ethical and Legal Concerns</td>
<td>20</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Ethical Concerns</td>
<td>20</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Legal Liability</td>
<td>21</td>
</tr>
<tr>
<td>3.3</td>
<td>Platforms, Hardware, and Software</td>
<td>22</td>
</tr>
<tr>
<td>3.4</td>
<td>Course Design Framework</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>Results</td>
<td>26</td>
</tr>
<tr>
<td>4.1</td>
<td>State of Ethical Hacking Education</td>
<td>26</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Analysis of NSF Funding Allocations and NCAE Certifications</td>
<td>26</td>
</tr>
<tr>
<td>4.1.2</td>
<td>A Survey of University Courses</td>
<td>27</td>
</tr>
<tr>
<td>4.1.2.1</td>
<td>Course Offering</td>
<td>28</td>
</tr>
<tr>
<td>4.1.2.2</td>
<td>Tooling Mentioned</td>
<td>28</td>
</tr>
<tr>
<td>4.2</td>
<td>Ethical and Legal Considerations</td>
<td>29</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Addressing Ethical Concerns</td>
<td>30</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Legal Concerns</td>
<td>32</td>
</tr>
<tr>
<td>4.3</td>
<td>Tool Compilation</td>
<td>33</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Platforms</td>
<td>33</td>
</tr>
<tr>
<td>4.3.1.1</td>
<td>Virtualization</td>
<td>34</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>5.1 State of Ethical Hacking Education</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>5.2 Course Development Framework</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>5.3 Other Difficulties</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>5.4 Conclusion</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>5.5 Future Work</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>APPENDICES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A Course Materials</td>
<td>81</td>
<td></td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 NSF Funding 2022 (in Millions of Dollars)</td>
<td>14</td>
</tr>
<tr>
<td>3.2 CAE Institutions as a Percentage of Computer Science Degree Granting Institutions in the US</td>
<td>16</td>
</tr>
<tr>
<td>3.3 University Security Course Offerings</td>
<td>17</td>
</tr>
<tr>
<td>3.4 Platforms Found in Publicly University Courses</td>
<td>18</td>
</tr>
<tr>
<td>3.5 Software Found in Publicly Available University Courses</td>
<td>19</td>
</tr>
<tr>
<td>4.1 Platforms Reference</td>
<td>33</td>
</tr>
<tr>
<td>4.2 Hardware Reference</td>
<td>38</td>
</tr>
<tr>
<td>4.3 Software Reference</td>
<td>46</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Flowchart to Help Determine what Platform(s) to Use</td>
<td>61</td>
</tr>
<tr>
<td>4.2</td>
<td>Flowchart to Help Determine what Hardware to Use</td>
<td>61</td>
</tr>
<tr>
<td>4.3</td>
<td>Flowchart to Help Determine what Software to Use</td>
<td>62</td>
</tr>
</tbody>
</table>
LIST OF ACRONYMS

CAE-C . . . Center of Academic Excellence in Cybersecurity

CAE-CD . . Center of Academic Excellence in Cyber Defense

CAE-CO . . Center of Academic Excellence in Cyber Operations

CAE-IAE . Center of Academic Excellence in Information Assurance Education

CAE-R . . . Center of Academic Excellence in Cyber Research

CISA . . . . Cybersecurity and Infrastructure Security Agency

CISE . . . . Computer and Information Science and Engineering

CSU . . . . . California State University

CTF . . . . . Capture the Flag

DHS . . . . . Department of Homeland Security

FBI . . . . . Federal Bureau of Investigation

IA . . . . . . Information Assurance

KU . . . . . Knowledge Unit

NCAE-C . . National Centers of Academic Excellence in Cybersecurity

NSA . . . . . National Security Agency

NSF . . . . . National Science Foundation

SaTC . . . . Secure and Trustworthy Cyberspace

SIGINT . . Signal Intelligence

UC . . . . . University of California
In the increasingly digital landscape, the need for digital applications and systems to be secure is self-evident. While designing defensive measures against known attacks is important, understanding an adversarial mindset and being able to consider future attacks and exploits that have not yet been executed is also an essential part of securing the digital landscape.

Ethical hacking is the application of offensive cybersecurity skills with an adversarial mindset to identify vulnerabilities and exploits without causing damage. These skills can be applied for a systematic system audit in a process called a penetration test, or pentest for short. While both of these terms may be used, for this paper I will primarily be using ethical hacking. Understanding the offensive techniques of cybersecurity is as important as understanding the defensive techniques, as it gives cybersecurity professionals the ability to be proactive in identifying potential vulnerabilities that may lead to exploits.

1.1 Problem

Ethical hacking is an effective way to test the security of a system, but it is often neglected when it comes to computer science education. While there is an abundance of research in discovering vulnerabilities and developing security tools, there is a rather limited amount of research in the pedagogy of ethical hacking. Educating people about hacking is a difficult problem for multiple reasons, some of which include the
ethical concerns of teaching real attacks [132, 105, 65] as well as properly setting up an environment that can be safely tested in [143].

While it is difficult to find concrete data on the exact numbers for cybercrime and the effects it has had, Sharif and Mohammed have conducted a literature review to estimate financial losses due to cybercrime, and it is evident that the general trend has been a year-over-year increase in financial losses and targeted attacks [75]. With this in mind, the need for more cybersecurity professionals who understand how to navigate the changing landscape of cybersecurity threats is paramount.

The need to strengthen the cybersecurity workforce is supported by the US government in the form of the National Centers of Academic Excellence in Cybersecurity (NCAE-C) Program [14, 87]. The NCAE-C program is sponsored by three governmental organizations: the Cybersecurity and Infrastructure Security Agency (CISA), the National Security Agency (NSA), and the Federal Bureau of Investigation (FBI). Its purpose is to serve as a way to accredit institutions that meet program requirements for teaching cybersecurity. Cybersecurity research efforts are also funded by the National Science Foundation (NSF) [95], specifically under the Secure and Trustworthy Cyberspace (SaTC) program [91]. While the accreditation and funding from these organizations support holistic cybersecurity education, they provide a useful litmus test of how much ethical hacking education is being supported at a university level.

Teaching ethical hacking first requires designing the courses to teach the different skills and tools necessary to conduct ethical hacking. To design a course, it is important to consider what infrastructure is necessary in order to do so. This is a notable gap in research in considering how to conduct a generalized approach to course design that is applicable across a broad range of ethical hacking skills. Therefore, in this work
I aim to create a framework to fill that gap as a resource for course topics, ethical modules, and tools and infrastructure.

1.2 Contribution

I offer two main contributions to this thesis: a survey into the current state of cybersecurity education and a framework for future educators to use to help with designing courses to teach ethical hacking skills.

- A survey into the current state of cybersecurity education
  - Examined data for universities accredited by the National Centers of Academic Excellence in Cybersecurity program;
  - Studied NSF funding for security-related grants;
  - Conducted specific case studies for a variety of universities course offerings and examined the publicly available resources for their security courses;

- Framework for creating ethical hacking courses
  - Proposed a process for utilizing the resources to create an ethical hacking course;
  - Designed a process to uphold ethical standards in ethical hacking courses;
  - Aggregated and organized a variety of platforms, software, and hardware;

1.3 Ethical Statement

In this work, I will be discussing a variety of software and hardware tools that can be used to exploit security vulnerabilities. However, the disclosure of the existence
of tools is not inherently unethical or dangerous because while these tools can be used in unethical ways, their existence is not a secret and their use is necessary for education. In this work, I also will not be discussing any specific vulnerabilities or exploits, and especially not any that have not been previously published or disclosed. Finally, in order to ensure that courses designed with the framework I have created will also uphold ethical standards, I have included a section on how to design ethical modules for ethical hacking courses.
Chapter 2

BACKGROUND AND MOTIVATION

In this chapter, we discuss the background information necessary to understand the rest of the work, as well as lay out the motivations and research questions guiding the work. In section 2.1, I discuss some of the institutions and organizations that contribute to improving cybersecurity education. Then in section 2.2, I discuss some terms that are not directly related to ethical hacking education, but will be referenced in this work. In section 2.3, I examine existing related academic work in my research area, and in section 2.4 I lay out my motivation and the research questions guiding this work.

Over the years, there has been a larger focus in teaching cybersecurity at an undergraduate level. The National Security Agency (NSA) launched the Center of Academic Excellence in Information Assurance Education (CAE-IAE) program in 1999 [87]. In 2015, only eighty institutions were designated by the NSA as CAE IA/CD [65], but in 2023 there were 426 with a CAE-C designation [90]. This increased focus on educating cybersecurity professionals is great, but the Knowledge Units (KUs) used to evaluate the CAE-Cs covers a wide variety of topics, and does not guarantee the teaching of ethical hacking skills. According to Li, less than 20% of the eighty institutions designated CAE IA/CD in 2015 offered a complete course on penetration testing [65].
2.1 Institutions for Cybersecurity Education

There are many organizations that provide resources for the development of cybersecurity curriculum. Notably, none of them are directly related to ethical hacking education specifically, but rather cybersecurity education as a whole. This is not a comprehensive view into what each organization provides. Instead, in this section I am only giving an overview of what each of these organizations as context to the rest of the work.

2.1.1 National Security Agency (NSA)

The National Security Agency was established in 1952 by Harry Truman with the missions of Signal Intelligence (SIGINT) and Information Assurance (IA). This broadly designates the monitoring of signals and prevention of unauthorized access to data as its primary duties. In this thesis, they are relevant as the main governmental organization sponsoring the Center of Academic Excellence in Cybersecurity program, with the Department of Homeland Security (DHS) as a cosponsor [87].

2.1.2 National Centers of Academic Excellence in Cybersecurity (NCAE-C)

The National Centers of Academic Excellence in Cybersecurity (NCAE-C) is a program that awards designations to universities in the US that meet accreditation requirements. Founded in 1999 [87], there are three designations: Center of Academic Excellence in Cyber Defense (CAE-CD), Center of Academic Excellence in Cyber Research (CAE-R), Center of Academic Excellence in Cyber Operations (CAE-CO). CAE-CD is the most widely awarded designation, and it can be awarded to any regionally accredited two-year, four-year, or graduate-level institutions in the United...
States offering cybersecurity degrees or certificates. CAE-R is a designation specifically for research, and only universities awarding Ph.D.s are eligible. CAE-CO is similar to CAE-CD, but it is required to be more technical, i.e. it must be computer science, computer engineering, and/or electrical engineering based, and must have opportunities for hands on learning. All designations must be reapplied for every 5 years.

2.1.3 National Science Foundation (NSF)

The National Science Foundation (NSF) is an organization created in 1950 in order to promote science. They primarily function by providing grants for research. The funding they provide represents about 25% of the federal funding provided to college and universities in the US. In the context of this paper, we primarily care about the Computer and Information Science and Engineering (CISE) category and the Secure and Trustworthy Cyberspace (SaTC) subcategory contained within it. These categories represent funding to computer science and cybersecurity research, respectively.

2.1.4 DeterLab

DeterLab [5] is a free network test bed funded by the NSF and the DHS. The hardware is hosted by UC Berkeley and USC ISI (University of Southern California Information Sciences Institute), and the software used to be based on the Emulab technology developed by the University of Utah, but it is now based on Merge [79]. It allows users to run experiments that can be configured via the DeterLab web interface on allocated physical machines that can be accessed via Secure Shell (SSH).

DeterLab is an isolated network from which no traffic can exit. All the traffic created by conducting experiments stays within the experimental network created for
it. Therefore, attacks can be conducted against machines in the network without affecting other users or the rest of the internet. DeterLab has also been used by a number of different universities for a number of different courses [76].

2.2 Terms

In this section, I will discuss some terms that will be used later in the thesis.

2.2.1 Capture the Flag

Capture the Flag (CTF) refers to a web based security competition which can be attack and defense based, or more commonly, jeopardy style. In this thesis, I will not be covering attack and defense CTFs, but rather will be primarily focusing on the challenge-based jeopardy style competitions where competitors can gain points by finding text strings by solving challenges.

2.2.2 White/Black Hat

Historically, the terms white-hat and black-hat hackers have been used to describe those with good and bad intentions, respectively. In an effort to utilize more inclusive and less ambiguous language, we will use the terms ethical and unethical instead.

2.3 Related Work

This section will discuss work done by other researchers that lay the groundwork for this thesis.
2.3.1 Tool Comparisons

Not many works exist that do a comparison of types of pentesting tools, though there is a paper by Islam and Fahmida that does some very detailed comparisons of open source wireless auditing tools [59], many of which like Wireshark and Kismet fall into the category of passive pentesting tools. It also has a Fig 7.6 which provides a table of similar style to this work, but comparison of some different tools. There is also a paper by Shebli and Beheshti that goes through the steps of pentesting and lists some of the tools used at each step, but does not do a comparison of the tools themselves [116].

2.3.2 Course Case Studies

There is a some substantial research that does specific dives into how to teach a specific course, and the learning outcomes of doing so. Trabelsi [131, 130, 132] has a number of papers published detailing the technical details of how to use certain tools to practice ethical hacking skills, and combines that with data analysis of student performance and course satisfaction. Another example is Penetration Testing Curriculum Development in Practice by Chengcheng Li [65]. In that paper, Li examines a course module structure of topics to teach with regard to a pentesting course. In 2018, Katz [61] compares the advantages of teaching ethical hacking with a focus on breadth versus depth, and in 2019, Dorofeev [25] proposes ways to teach an ethical hacking methodology rather than only the technical skills. With the 2020 COVID pandemic, another major development came with papers like one by Wang [143] exploring how to set up an effective teaching environment for teaching security testing. OConnor and Strickland [97] has work in the same vein teaching a mobile and wireless
security course, while detailing the course infrastructure used and some details about how students performed on a semester-long CTF.

2.3.3 Developing Lab Environments

In recent years, there has begun to be more emphasis on teaching these skills in an authentic environment with more realistic lab environments, and even some work allowing students to test actual university applications [125]. Yet, little research exists to show whether these more authentic environments serve to better educate students. In fact, perhaps due to the COVID-19 pandemic, many papers in recent years have been focusing on how labs can be virtualized [148, 52]. At Cal Poly San Luis Obispo, few computer science classes require any kind of hardware beyond a computer. This is fairly reasonable for most classes, but some topics may be more difficult to recreate in a virtualized environment. For example, classes about networking, both wired and wireless, are still taught with hardware components.

2.3.4 Ethical Concerns

While conducting cybersecurity education, there are a number of ethical questions that are raised. In 2013 Pike et al. [105] published a paper addressing the topic, and it primarily focuses on the ethics of teaching real attack techniques. Another paper in 2013 by Trabelsi [132] is an interesting case study that notes that universities tend to see an increase in the detection of a specific attack, in this case DoS (Denial of Service), when the course teaching DoS is taught. In 2015, [49] came to the conclusion that most educators and security professionals believe that the positive points of teaching ethical hacking outweigh the negatives. Due to the inherent risks
of students misapplying skills learned in ethical hacking courses, it is important to teach students some basic ethics.

2.4 Motivation

Part of the motivation for this work is increasing amount of high profile cybercrime, which indicates a need for more security professionals in industry. The ease of access to powerful security tools is also a motivating factor, as it is important to educate people on how to use these tools in an ethical manner. While there is much available research with regard to security and new exploits, there is less work available on how to effectively teach students to understand how attackers can utilize vulnerabilities and turn them into exploits, i.e. ethical hacking. This work seeks to serve as a resource for educators of the future generations of cybersecurity experts by providing a variety of information about ways to develop many different types of ethical hacking courses.

This thesis seeks to answer three main research questions:

1. What is the current state of teaching ethical hacking at a university level?

2. How do you address legal and ethical concerns with teaching a set of skills that can be used for criminal purposes?

3. What resources can be leveraged in order to teach ethical hacking?
Chapter 3

METHODOLOGY

In this chapter, we will discuss the methodology of our research and how we sought to answer each of the research questions posed in Section 2.4. Section 3.1 focuses on examining the current state of cybersecurity education in the United States at an undergraduate level by first looking at funding and certification programs in order to develop an understanding of the support the NSF and NSA provide for cybersecurity education and research. I also examined fifteen universities to examine what courses they offer and what platforms, hardware, and software those courses use. Section 3.2 covers what ethical and legal concerns need to be addressed when trying to teach these skills. Then, in section 3.3 we take a look at what resources will be necessary in order to conduct a course in ethical hacking. In section 3.4 of this chapter, we will discuss how and why the elements of the framework were chosen. By combining the answers to the three research questions, I have formulated a framework that can be utilized to create a course for ethical hacking that satisfies certification requirements, is conducted responsibly, and leverages resources appropriate to the use case.

3.1 University Education

Investigating the first research question consisted of conducting research into aggregate data about cybersecurity education and a survey of specific universities and their cybersecurity course offerings.
3.1.1 Aggregate Data

It is difficult to gather exact metrics on the aggregate level of security education in the US higher education system because there is no simple metric by which to judge whether an institution teaches cybersecurity. However, it is possible to gain some level of insight by looking into funding and certification programs.

For a more quantifiable view of cybersecurity education in the US, I looked at two main sources. First was the National Science Foundation (NSF) data for awards to computer science/engineering and cybersecurity research [95, 93]. Second, I also looked at the National Centers of Academic Excellence in Cybersecurity (NCAE-C) program [87] created by the National Security Agency (NSA) and other governmental partners that certifies universities for teaching or researching cybersecurity.

3.1.1.1 NSF Funding

The NSF is a US government organization that provides funding for science and technology research. By looking at the funding provided for different fields and programs, we can get a sense of how well-supported cybersecurity is, at least from a financial perspective. The NSF publishes data sets of all the funding awards they provide in an online database called “NSF by the numbers” [94] and through that we can figure out how much funding is being provided for different categories of the NSF.

The NSF has an award obligation in 2022 in the category of Computer and Information Science and Engineering (CISE) for a total of $1,015.57M, which is just over 1 billion dollars [92]. This is out of $8,838.00M total award obligation for 2022 [92], which is about 11.49% of the total funding. There are 9 total categories that NSF
<table>
<thead>
<tr>
<th></th>
<th>Funding Granted</th>
<th>Percentage of Whole</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISE</td>
<td>$1,015.57M</td>
<td>11.49%</td>
</tr>
<tr>
<td>SaTC</td>
<td>$396.00M</td>
<td>4.48%</td>
</tr>
<tr>
<td>All Awards</td>
<td>$8,838.00M</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

awards funding to in addition to a variety of overhead costs, so the CISE category took a rather proportionate, if not slightly high, amount of funding in the year 2022.

Of all the money allocated to CISE in 2022, $396 million out of approximately $1 billion allocated towards CISE were allocated towards Secure and Trustworthy Cyberspace (SaTC) [93]. This data is found by downloading all the SaTC awards from the NSF award search database. This represents a significant emphasis on research in cybersecurity by the NSF.

However, it is important to note that only some amount of this funding is going towards pedagogical research, and even when it’s going towards education, it may not be specifically going towards ethical hacking education.

3.1.1.2 CAE

The NSA runs a program called National Centers of Academic Excellence in Cybersecurity where it accredits universities for cybersecurity education. We can use data from this accreditation program in order to help us gauge the current state of cybersecurity education. According to Li [65], in 2015, there were eighty institutions designated as Centers for Academic Excellence in Information Assurance/Cyber Defense (CAE IA/CD). As of time of writing (2023), the name has since dropped the IA for information assurance, and instead is simply referred to as CAE-CD. Additionally, there are also two additional designations for cyber research and cyber operations, Center of Academic Excellence in Cyber Research (CAE-R) and Center of Academic
Excellence in Cyber Operations (CAE-CO) respectively. CAE-R is a designation that focuses on doctoral research and is only awarded to schools with doctoral programs and DoD schools [87]. CAE-CO is more like the original CAE-CD designation, but it is more technical and requires opportunities for hands-on labs and applications. Collectively, they are referred to as Center of Academic Excellence in Cybersecurity (CAE-C). According to the CAE-C community map website [90], at there are now 426 institutions with any kind of CAE-C designation, 388 with a CAE-CD, 78 with a CAE-R, and 20 with a CAE-CO. Note that institutions can hold several designations simultaneously, hence the sum of the designations not summing up to 426.

Note that according to Li, only twenty percent of the eighty CAE-IA/CD designated institutions in 2015 offered “a complete course on pentest” [65]. It is unclear exactly how this percentage was determined as the website cited no longer works without a Department of Defense root certificate, and the web archived version of the page does not make it clear on how this figure was determined. It is also worth noting that the Knowledge Units (KUs) defined at the time of writing for CAE-CD [89] include an optional unit for penetration testing. However, it is unclear what the relative frequency of optional knowledge units being implemented is. If we assume that the percentage has stayed the same since 2015, we could speculate that around eighty-five institutions in the US that are accredited with CAE-CD have a complete course on penetration testing. For a sense of scale, according to data USA [22], there are 1135 institutions in the US that have a computer science degree program [21]. This means approximately 38% of institutions in the US that have a computer science degree are accredited with either CAE-CD, CAE-R, or CAE-CO. We can also speculate that around 8% of institutions in the US are accredited and have a complete course on penetration testing.
Table 3.2: CAE Institutions as a Percentage of Computer Science Degree Granting Institutions in the US

<table>
<thead>
<tr>
<th></th>
<th>Number of Institutions</th>
<th>% of CAE Institutions</th>
<th>% of all Institutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAE-CD</td>
<td>388</td>
<td>91.08%</td>
<td>34.19%</td>
</tr>
<tr>
<td>CAE-R</td>
<td>78</td>
<td>18.31%</td>
<td>6.87%</td>
</tr>
<tr>
<td>CAE-CO</td>
<td>20</td>
<td>4.69%</td>
<td>1.76%</td>
</tr>
<tr>
<td>Any</td>
<td>426</td>
<td>100.00%</td>
<td>37.53%</td>
</tr>
</tbody>
</table>

There are of course some important caveats to this data. Firstly, the NCAE-C accreditation program is voluntary, so it does not represent all the universities in the US that teach cybersecurity, only those that choose to be accredited. Secondly, because the knowledge unit for penetration testing is optional, it is unclear what percentage of CAE-CD institutions implement a penetration testing course.

3.1.2 Survey of University Courses

In this section, I picked fifteen universities and manually examined their course offerings in order to gain a more granular understanding of what kind of cybersecurity courses are offered in US universities. Because looking at the funding and certification data gives only a broad overview of how security is funded and certified and does not split between ethical hacking education and other types of cybersecurity education, this more granular look is required. In addition, by looking at the publicly available resources for the cybersecurity courses, I also was able to investigate some of the platforms, software, and hardware that I could reference for section 3.3.

The universities in the study were chosen to represent a variety of types of universities such as universities from the University of California (UC) system, universities from the California State University (CSU) system, public universities not in California, and private universities.
Table 3.3: University Security Course Offerings

<table>
<thead>
<tr>
<th></th>
<th>Intro to Security</th>
<th>System</th>
<th>Network</th>
<th>Hardware</th>
<th>Cryptography</th>
<th>Web</th>
<th>Penetration Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC Berkeley</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UC Los Angeles</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC San Diego</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cal Poly SLO</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Cal Poly Pomona</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>San Jose State</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Georgia Tech</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UIUC</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arizona State</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Carnegie Mellon</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MIT</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Caltech</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stanford</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cornell</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I first chose to look at Cal Poly Pomona and SJSU as they are both well known computer science universities in the CSU system, but neither had course material that was publicly accessible. I then looked at UC system schools, since they are also public universities in California. In order to expand my survey coverage beyond California, I picked a number of schools that rank highly for computer science education. Additionally, I selected some universities because they had significant amounts of publicly availability of course material. I also selected some universities because of external references to their material, such as Stanford’s Zoobar assignment being used in several universities outside of Stanford. The selection does have a bias towards Californian schools, as I am conducting this research from Cal Poly SLO.

Surprisingly, there were a number of universities that only offered 3 or fewer cybersecurity courses such as UCLA, Caltech, MIT, and Cornell. Some universities like Georgia Tech also offer far more graduate level computer security classes rather than undergraduate level classes. Out of all the universities examined, only one, Carnegie Mellon, has a course specifically on penetration testing. Table 3.3 gives an overview of what types of security courses I was able to find listed for each of the universities.
at the time of writing. Additionally, my information for Cal Poly SLO is insider knowledge since I have access to the canvas courses of classes I have personally taken and did not have similar access for other schools.

**Table 3.4: Platforms Found in Publicly University Courses**

<table>
<thead>
<tr>
<th>University</th>
<th>DETER</th>
<th>Virtualbox/VMware</th>
<th>Docker</th>
<th>TryHackMe</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC Berkeley</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC Los Angeles</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC San Diego</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cal Poly SLO(^a)</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>San Jose State</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Cal Poly Pomona(^a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Georgia Tech</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UIUC(^a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arizona State(^a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carnegie Mellon</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>MIT</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Caltech(^a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stanford</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Cornell</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

\(^a\) No public course websites found for this university
Table 3.5: Software Found in Publicly Available University Courses

<table>
<thead>
<tr>
<th>University</th>
<th>Hydra</th>
<th>Wireshark</th>
<th>Nmap</th>
<th>GDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC Berkeley</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC Los Angeles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC San Diego</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cal Poly SLO(^a)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>San Jose State</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cal Poly Pomona(^a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Georgia Tech</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UIUC(^a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arizona State(^a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carnegie Mellon</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>MIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Caltech(^a)</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Stanford</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Cornell</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

\(^a\) No public course websites found for this university

Looking into the information available for each course, it varied greatly depending on the university and class. While some universities such as Stanford, UC Berkeley, and Cornell have publicly available course information, others like ASU, Cal Poly Pomona, and SJSU do not. In Table 3.4 and Table 3.5 I compiled the platforms and software utilized by the courses at each university based on what course information I
could find online. Thus, keep in mind that an empty cell does not mean that a certain platform or tool is not utilized by the university in their cybersecurity courses, as it may be used in a course that does not have publicly available course information. Full references for all course websites is found in appendix A.

3.2 Ethical and Legal Concerns

To answer the second research question, I first needed to identify what ethical and legal concerns come up when teaching ethical hacking. Based on my background research, it is clear that there is a substantial probability that students will apply the skills they learn in ethical hacking courses in contexts that they should not. One example is a case study by Trabelsi where 85% of students responded to an anonymous survey saying that they attempted denial-of-service attacks outside the classroom environment [132]. Therefore, it is important to consider how to minimize these occurrences and to understand what kind of liability an educator holds towards their students in the case where a student does cause a problem.

In this chapter I will only look at the concerns, but in the next chapter I will discuss how these concerns can be addressed.

3.2.1 Ethical Concerns

Ethical concerns begin in the classroom in the very environments where students learn ethical hacking skills. For labs where skills are taught, it is important that the targets are not live applications or in use infrastructure that may affect the community’s access. As such, the configuration of these environments is an essential part in upholding ethical standards for teaching ethical hacking skills.
Additionally, beyond the classroom, there is the additional possibility of students utilizing the skills they have learned to unethically exploit vulnerabilities in systems they do not have permission to exploit. While it is near impossible to fully prevent this from happening, by educating students on the ethics side of ethical hacking, it may be possible to help mitigate its occurrence. There has been discussion in the past about whether teaching the skills of offensive security is unethical in of itself, such as the survey of information security professionals conducted by Pike [105]. Within the work, Pike finds that there is universal support for teaching ethical hacking at universities in cybersecurity programs, but with important caveats and recommendations to help reduce the likelihood of unethical behavior.

In the following chapter, I will discuss how I address these concerns in the course design framework.

3.2.2 Legal Liability

In addition to ethical concerns, it is also important to consider what level of legal liability teachers have for the actions taken by their students. In one paper, Trabelsi reported that the average number of invalid ARP packets injected into the university network consistently increased in the days following the lab exercises for ARP poisoning [130]. In the same paper, Trabelsi surveyed students taking the class and they self reported using DoS and MitM attacks outside the isolated lab environment. Therefore, it’s important to figure out to what extent these actions could put liability on an educator or the university.

As a legal disclaimer, I am not a lawyer and this section has not been reviewed by a lawyer, so this should be taken purely as research and a jumping off point and not legal advice. While there is little specific literature on the exact topic of legal
liability for teaching ethical hacking, there is existing work like that by Perez-Velez that discusses the general personal liability of student affairs professionals [103]. In the paper, Perez-Velez discusses some of the liability educators might face and notes the two main categories of criminal and civil liability. Criminal liability is not likely to come up unless the educator is somehow assisting the criminal action, as it requires the educator to have criminal intent. Within civil liability, I think the most crucial subcategory to consider is tort liability. Under tort liability, educators can be found liable for negligence if said negligence or a failure to act causes a plaintiff significant harm.

Another concern about legality also rests with student understanding of the legality of their actions. Within a court of law, lack of understanding of the law is not a valid defense for violating the law [103]. As educators, if we are to teach students skills that have a reasonably low barrier to intentional or accidental illegal usage, we should also make sure students are aware of what kind of laws they may find themselves in violation of.

3.3 Platforms, Hardware, and Software

To answer the third research question, I conducted a survey of what resources were available for the purposes of teaching ethical hacking. The third research question is broad since there are many technical elements that go into setting up a course, so I categorized the technical resources into platforms, hardware, and software. In order to conduct my survey, I noted resources utilized in the publicly available course information from my university survey, papers about specific course designs, as well as online resources designed to help would be penetration testers.
I designated platforms as its own category because in making sure that a course is taught ethically, it is essential that attacks students conduct are executed in properly sand-boxed environments that will not impact existing infrastructure or other students. I categorized hardware as a separate attack resource because a limitation in hardware accessibility may severely limit the ability of a course to run smoothly if certain types of attacks are to be conducted, and so they should be considered before software. However, the first consideration should still be platforms, as they will determine the overall infrastructure of a course. Finally, software tends to be the most flexible resources as there are many options to achieve many of the same types of results, so it is considered last.

I decided to split the resources for designing a course in this manner for a couple of reasons. It helps to categorize all the available resources, making the information about available resources more digestible. Additionally, I suspect that often times the limiting factor in designing a course will be either platform or hardware limitations. Therefore, I wanted to focus specifically on alternatives and options to tailoring courses to available resources.

In the following list, I define each category to clarify what kind of resources will fall into which category.

1. **Platforms:** Platforms refers to the underlying infrastructure of the course and resources needed to configure and set up the lab environment. Platforms merely provide a mean to design environments, and students learning to interact with platforms is not typically a learning objective. This is an important distinction because when designing a course, it is preferable that less time is wasted with students struggling with configuration issues so that more time can be spent on using the tools that are aligned with the learning objectives. In general, I
categorized resources that enabled the creation of an environment to conduct lab activities as platforms. Some examples are virtualization software and web platforms.

2. **Hardware**: Hardware refers to physical devices that are necessary to conduct attacks. While much can be learned through virtualized and purely software environments, there are some attacks that do require the usage of hardware, such as wireless sniffing and spoofing. Utilizing hardware in a course comes with an inherent storage requirement, which may be a drawback in some cases, for example, if one needs to share lab space. Hardware also typically adds to the cost of running a course, making the usage of hardware potentially infeasible depending on budget.

3. **Software**: Software refers to software applications or tools whose usage in conducting attacks will advance student understanding of the learning objectives. This is most likely the biggest consideration when designing a course, as allowing students to gain familiarity with tools that can be used in penetration testing will better equip them as cybersecurity professionals. When surveying the different software available, I considered not only their functionality and purpose, but also their availability, maintenance, and popularity.

### 3.4 Course Design Framework

Finally, as the other main contribution of this work, I designed a framework that would enable educators to design a meaningful ethical hacking course. To do so requires synthesizing the ethical and legal concerns with the knowledge of the current state of cybersecurity education and the platforms, hardware, and software available
to teach ethical hacking. From there, I created a step-by-step process to develop all the necessities for an ethical hacking course.

Components of the framework include a list of possible topics including topics from the NCAE-C penetration testing KU, recommendation for ways to reinforce ethical behavior, and flowcharts to help determine what platforms, hardware, and software to use.
In this chapter, I discuss the results of conducting my research in line with the methodology outlined in chapter 3. In section 4.1, I will discuss the implications of the aggregate data and the survey result of university courses. Section 4.2 discusses how the ethical and legal concerns can be addressed. Then in section 4.3 the tables of available platforms, hardware, and software are presented with brief descriptions of each. Finally, in section 4.4, I propose my framework for designing ethical hacking courses which incorporates the tables in section 4.3.

4.1 State of Ethical Hacking Education

This section is to analyze the data from section 3.1 to understand what it implies about the current state of ethical hacking education.

4.1.1 Analysis of NSF Funding Allocations and NCAE Certifications

Considering the total amount of funding being provided to the NSF, the amount of funding for CISE is quite a large proportion. However, it is also not the largest quantity of funding provided, as it trails behind Geosciences (GEO) and Mathematical and Physical Sciences (MPS) [92]. Furthermore, the funding for CISE is further subdivided into other categories, which means 40% of the funding for CISE is specifically for SaTC. Even if SaTC does not necessarily represent funding for teaching and edu-
cation of cybersecurity nor ethical hacking specifically, it still shows a vested interest from the NSF in the advancement of the field of cybersecurity.

Additionally, the NSA provides a NCAE-C certification to universities that meet its requirements for teaching cybersecurity. At the time of writing, there are now 426 institutions with some kind of NCAE-C certification. This represents a more than five-fold increase in the number of certified institutions, which suggests that universities are demonstrating more interest in teaching cybersecurity and getting certified for teaching cybersecurity.

Overall, the aggregate data suggests that there has been an increase in the interest in teaching cybersecurity at an undergraduate level, though there is still plenty of room to grow. It also shows that the growth in cybersecurity education is relatively recent. This combination of factors contributed to the motivation for creating this thesis, as it suggests that a framework for developing ethical hacking courses is likely to be useful.

### 4.1.2 A Survey of University Courses

From my study of universities and their courses for security, there are two distinct takeaways. One is a better understanding of what kind of courses universities are offering in cybersecurity and ethical hacking skills. The other is a baseline understanding of some of the kinds of tools that are currently in use at universities, which I was able to gain through the publicly available course information for each school.
4.1.2.1 Course Offering

Looking at table 3.3 there are several interesting trends to notice. The most commonly offered class is cryptography, which is somewhat unexpected, as there were a few universities that did not offer any broad introductory courses, but did offer cryptography. A close second are intro to security courses, which is inline with logical expectations since it covers the broadest amount of topics and is applicable to the largest number of students. Then, there is a second group of classes that have a reasonably large number of schools offering them: system security and network security. Finally, we have hardware security, web security, and penetration testing. These courses are the most niche, and are not as popularly offered. While this is by no means an exhaustive study of all the course offerings from universities, it at least offers a glimpse into what the distribution of cybersecurity topics that are taught. From this, we can glean that offensive security and ethical hacking skills are not currently at the forefront of security education, further motivating this thesis.

4.1.2.2 Tooling Mentioned

In tables 3.4 and 3.5, I examined the software and platforms that were utilized by each university in whatever publicly available course information they had. While this has the obvious downside of being limited by courses that happen to have a public website, due to time and scope constraints, it was infeasible to individually reach out to every course instructor to gain access to courses where it was not publicly available.

What I was able to find was that the most commonly used platform was some kind of virtualization like VirtualBox or VMware. Their flexibility leads to them being able to be utilized in a variety of situations and topics. Docker was another interesting vir-
virtualization platform that could be used in order to configure vulnerable applications, but it may require a significant amount of instructor effort to set it up.

The other very popular kind of platform is a web capture the flag (CTF). While not self-evident from looking at table 3.4, there were many courses that created and published their own web CTF challenges. In table 3.4, I specifically note tryhackme.com because it is not a self-hosted ctf, rather it is a commercial product that partners with educational institutions. This led to me investigating what other web CTF solutions existed, in section 4.3.

On the software side, it was a lot harder to find a clear pattern. The kind of software utilized depended largely on the use case. Nmap was used quite a few times, which makes sense as a well-supported and reputable reconnaissance tool, but other than that, I did not find any significant patterns from my survey here. To reiterate my point from the prior section, this does not imply that the universities surveyed do not utilize these software tools in the courses, only that I did not find their usage while viewing the publicly available course material. Because of a lack of a clear pattern in the software found, I had to search broadly when I began looking at software tools in section 4.3.

4.2 Ethical and Legal Considerations

In this section, I will discuss some of the possible strategies and mitigations that can be put in place to help keep the course as ethically run as possible. Additionally, I will discuss important subjects to cover to limit the amount of legal trouble that instructors and students can find themselves.
4.2.1 Addressing Ethical Concerns

Without the consideration of the ethical aspects of hacking, hacking becomes a potentially dangerous and harmful skill. However, this is not to say that hacking should not be taught, as a survey of 206 information security professionals by Pike in 2013 [105] found unanimous support for the teaching of ethical hacking in university level cybersecurity courses. In this work, Pike recognized several categories of recommendations on how to help students learn and adhere to ethical principles, including the following four that were recognized by 80(39%) respondents or more:

- Social interaction/support system
- Competition
- Recognition
- Ongoing skills development

From Pike’s work, I think it is particularly useful to highlight the two propositions that he suggests related to the recommendations for recognition and ongoing skills development.

For recognition, Pike proposes, ”Providing recognition for [ethical] hacking activities that are relevant to an individual’s peer group will reduce the likelihood that a student will need to engage in [unethical] activities to gain recognition and will reinforce the value of the [ethical] activity” [105]. In practice, this means that students need to have some way of being recognized by their peers for utilizing their skills in hacking ethically.

For ongoing skills development, Pike states, ”Providing challenges that are properly matched with students’ skill levels will serve to reduce the likelihood that students
will turn to black-hat activities” [105]. This is also an important consideration to take into account, since it is inevitable for there to be some level of disparity between the skill levels of individual students, so having one uniform level of challenge may make the course too easy for some, and too difficult for others.

One way to address both of the propositions at once is through the usage of asynchronous CTFs that are run throughout the quarter. This allows students to both get recognition for their completion of challenges, while also providing a scalable level of difficulty that students can self-impose. This has been done before in OConnor and Stricklan’s Mobile and Wireless security course [97] and this kind of approach in general has been shown to reveal strong correlations between students who perform poorly in these kinds of activities and their overall marks [13]. Therefore, it is both a useful approach in matching Pike’s propositions to learn and develop ethical skills and in pedagogical evaluation of students’ success. Of course, this is not the only way to address Pike’s propositions, and there are many other options such as team-based penetration tests of live applications [125].

While using motivating factors and class structure to keep students’ behavior ethical is important, it is also important to formally recognize some of the ethical problems that come into play with doing ethical hacking. A simple but important way to do is having a code of ethics. While you can always develop your own code of ethics for a course, another easier strategy can be to utilize existing codes by computing organizations. One example of this is the Association for Computing Machinery (ACM) has a code of ethics and professional conduct [37]. In the first section of the code of ethics, it describes the general ethical principles that computing professionals should adhere to. You can directly reference their code of ethics, or use it as a jumping off point in creating a code of conduct for a course.
Having a code of ethics is an important step towards minimizing ambiguity of what is considered ethical and unethical, but it is also important to consider the timing and context of when reminders of appropriate ethical behaviors are given. Cook [16] suggests warning students about capabilities of potentially dangerous tools before they are taught how to use them, or encouraging informed discussion about ethical behavior after students have had some experience with the course material. Something that Cook also found in his non-scientific interviews with security educators was that the signing of ethical honor codes was a common practice, and while it is hard to determine how effective these are in the context of ethical hacking, honor codes have precedent in being used for academic integrity purposes and have seen some improvement in the perception of the class and instructor [63].

4.2.2 Legal Concerns

Besides ethical concerns, there are also potential legal ramifications of not properly conveying the potential risks of improper behavior in an ethical hacking course. It is not uncommon for students to not read university policies about computer use [49], so it is important to list them in your course, like Bates does in his introduction to computer security course [1]. Bates also cites the computer fraud and abuse act (CFAA) which provides an important legal context to what activities are legal for students to engage in. Additionally, explaining what the potential consequences (inability to complete a security clearance, etc.) of inappropriate behavior are may also serve as an effective deterrent [16].
### Table 4.1: Platforms Reference

<table>
<thead>
<tr>
<th></th>
<th>Commercial License</th>
<th>Open Source</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Virtualization</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Docker</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Virtualbox</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>VMware</td>
<td>Depends on OS</td>
<td>No</td>
</tr>
<tr>
<td><strong>Web CTFs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overthewire.org</td>
<td>No</td>
<td>Website accepts pull requests</td>
</tr>
<tr>
<td>Microcorruption.com</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Tryhackme.com</td>
<td>Some features</td>
<td>No</td>
</tr>
<tr>
<td>Pwn.college</td>
<td>No</td>
<td>Infrastructure yes, not all modules though</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DETER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Stanford CS155 Zoobar</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Prebuilt VMs/Disk Images</td>
<td>No</td>
<td>Depends</td>
</tr>
<tr>
<td>Google Colab</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

### 4.3 Tool Compilation

I have divided the different aspects of an ethical hacking environment into three main categories: platforms, software, and hardware. Through my research, I have compiled the following non-exhaustive list of resources in each category. By utilizing these resources, educators and researchers will be able to more easily create new environments for ethical hacking education.

#### 4.3.1 Platforms

Lab environments need to be hosted somewhere, and the form this takes can vary wildly depending on the use case. Platforms vary from per-student deployments to distributed networks for testing. Some platforms are relatively bare bones, whereas others provide more of an all-in-one solution for course development.
4.3.1.1 Virtualization

Virtualization is a common way to host a security based courses in university classes on security, and for good reason. Virtualization allows for fine-grained control over the environment which students will interact with. It also is a good way to isolate potential damage caused by interacting with various ethical hacking tools or malware samples. Utilizing containers or virtual machine (VM) images also means that students will run consistent environments that can be configured with whatever tools are necessary for a course, and those tools can be verified to be compatible with just one original instance of a container or VM image.

- **Docker**: Docker is a containerization platform that allows for the creation of containers using a disk image as a starting point [24]. This is lightweight and does not require the virtualization of a kernel, as it utilizes the same kernel as the underlying operating system and therefore does not need to virtualize its own. Docker can be a good choice if you want to configure a remote server for students to access, and you need to spin up many similar instances. If complex networking is required, you may run into some limitations.

- **VirtualBox**: VirtualBox is an open-source virtual machine platform developed by Oracle that can be used to run virtual machine images [99]. It is free and open source, which makes it easy for students to download and use. It is also supported by tools like Vagrant [51] which can make configuring multiple machines and networking between them easier.

- **VMware**: VMware has a commercial suite of virtual machine software [142], some of which needs licensing that may or may not be provided through the university. Different versions of its software may be needed for different operating systems, i.e. macOS requires VMware Fusion. VMware also has some cloud
based solutions such as vSphere that allow for a network to be modeled, which while not suitable for individual students to configure, can provide a vulnerable environment for teaching ethical hacking skills while not exposing any real university infrastructure to attacks.

4.3.1.2 Web CTFs

Capture the Flag (CTF) competitions are a common type of security competition that can occur live or asynchronously, where teams compete to find flags for points. A common style of CTF is jeopardy style, where a variety of security challenges are available for points. Many live CTFs leave their challenges available for others to use even after the event has concluded. This can provide a useful resource for educators to give to students if the challenges are relevant to the class material, and can help prepare students who are interested in participating in CTFs in the future. There are also a variety of academic pedagogical research around the infrastructure in hosting your own CTF, such as the work by Trickel [133].

- Overthewire.com: Overthewire provides a variety of CTFs which cover a wide variety of difficulties [124]. While the website is online, the challenges are accessed by opening an ssh connection into the provided server. While this requires a basic understanding of ssh, it is not a very high overhead cost for students.

- Microcorruption.com: Microcorruption is a CTF focused more on assembly level exploits, backed by the NCC group and Square Inc [83]. This can be useful for classes dealing with microprocessor exploits. It does require somewhat linear progression, which can be a drawback if only certain challenges are relevant to the course material.
• **Tryhackme.com**: Tryhackme is a commercial website that offers a variety of modules to teach cybersecurity skills [134]. It is utilized by CMU in their 95-483 & 95-883 Ethical Penetration Testing course [60]. It can be used purely as a web resource for students to access, but there are some integrations that are offered for educators. For example, you can monitor student progress, put them into groups, etc., but this does require a paid plan to use.

• **Pwn.college**: Pwn.college is a website run by the University of Arizona that offers a wide variety of challenges [117]. Pwn.college operates by providing each student with a unique virtual desktop and text editor within the browser that the students can use to solve each of the challenges. This makes it easier to teach students how to use applications like Ghidra, ANGR, etc. without necessarily needing to have those applications set up on their own devices. They also welcome educators to request private instances of the infrastructure. It is based on the CTFd framework, but their infrastructure is also open source.

4.3.1.3 Other

There are also a variety of other platforms that can be utilized to create a lab environment for students. The platforms in this section are a mix of multiple different options that may be relevant depending on what kind of course you are developing.

• **Prebuilt VMs/Disk Images**: While not platforms in of themselves, there have been several operating systems developed with the express purpose of performing security exploits. This can be useful if you are developing virtual machine images or remote servers. Some examples of this include Kali, Security Onion, and Backbox which are general use VMs with security tools pre-installed.
There are also VMs that are designed to be exploited like Metasploitable 2, DE-ICE, and OWASP VM [65].

- **DETER**: As previously described in section 2.1.4, DETER is a collaboration of several universities and the department of homeland security that allows for a safe environment in order to test network based attacks[5]. It is still currently in use, however, now a new test bed called SPHERE whose development is funded by NSF is being released [77].

- **Stanford CS155**: Stanford’s CS155 class utilizes docker to run a vulnerable web application, called bitbar. There are vulnerabilities built into the application so that students can learn how to exploit different web vulnerabilities [19]. MIT also uses this application in its computer systems security [85] and the web server application used for Cornell’s CS 5435 class: Security and Privacy Concepts in the Wild [141], is inspired by both. Using the application or one of its derivatives may be useful for teaching web-based attacks.

- **Google Colab**: "Colab is a hosted Jupyter Notebook service" [36] which can be used for a variety of Python-related teaching modules. This can be useful for teaching skills that require a high prerequisite knowledge of Python or certain dependencies like Keras or TensorFlow. For example, Google hosts a proof of concept for adversarial examples using the fast gradient signed method (FGSM) originally shown in work by Goodfellow [129, 35].

- **Physical Labs**: Creating physical labs will be highly dependent on the availability of lab space to hold hardware. However, in some situations, physical lab environments may be useful to offer practical hands-on experience. This is most feasible in small classroom environments, e.g. upper division classes. Examining certain ethical hacking tools and skills may also be easier to do in a physical lab environment, e.g. rubber duckies or wireless packet sniffing.
## Table 4.2: Hardware Reference

<table>
<thead>
<tr>
<th>Price</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generic Software Defined Radio (SDR)</strong></td>
<td></td>
</tr>
<tr>
<td>RTL-SDR Dongle</td>
<td>$30 500 kHz - 1.75 GHz</td>
</tr>
<tr>
<td>Panda WiFi 802.11 Dongle</td>
<td>$30-50 Wi-Fi (802.11)</td>
</tr>
<tr>
<td>HackRF SDR</td>
<td>$300-400 Half-Duplex 1 MHz - 6 GHz</td>
</tr>
<tr>
<td>BladeRF SDR</td>
<td>$540 Full-Duplex 300MHz - 3.8GHz</td>
</tr>
<tr>
<td><strong>Protocol Specific SDR</strong></td>
<td></td>
</tr>
<tr>
<td>Attify Mote</td>
<td>$149-190 KillerBee/Zigbee</td>
</tr>
<tr>
<td>Yardstick One</td>
<td>$100-150 KillerZee/Zigbee</td>
</tr>
<tr>
<td>Ubertooth One</td>
<td>$100 Bluetooth/BLE</td>
</tr>
<tr>
<td>nRF52840 Dongle</td>
<td>$15 Bluetooth/BLE</td>
</tr>
<tr>
<td>CrazyRadio Dongle</td>
<td>$35 MouseJack (RF)</td>
</tr>
<tr>
<td>USRP N210</td>
<td>$3,300 Full-Duplex DC - 6 Ghz</td>
</tr>
<tr>
<td><strong>Hak5 Hardware</strong></td>
<td></td>
</tr>
<tr>
<td>WiFi Pineapple Mark VII</td>
<td>$120+ Wi-Fi (802.11)</td>
</tr>
<tr>
<td>USB Rubber Ducky</td>
<td>$80 USB</td>
</tr>
<tr>
<td>Bash Bunny Mark II</td>
<td>$120 USB</td>
</tr>
<tr>
<td>LAN Turtle SD</td>
<td>$80 USB-A to Ethernet</td>
</tr>
<tr>
<td>Key Croc</td>
<td>$120 USB</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td></td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>$80 General Use</td>
</tr>
<tr>
<td>Flipper Zero</td>
<td>$170 RFID, Sub 1Ghz, NFC, Bluetooth, Infrared</td>
</tr>
<tr>
<td>Pwnagatchi</td>
<td>$70-80 Wi-Fi (802.11)</td>
</tr>
<tr>
<td>HydraBus</td>
<td>$110 Embedded Hardware, NFC</td>
</tr>
<tr>
<td>Proxmark</td>
<td>$340 RFID</td>
</tr>
</tbody>
</table>

### 4.3.2 Hardware

Hardware is often limited by both space and budget, but some tools and skills may require the use of hardware or are generally difficult to virtualize or simulate. There are also many hardware hacking skills that can lead to other exploits.
4.3.2.1 Generic Software Defined Radio (SDR)

Many of these hardware options for SDR are cited from Table 2 of O'Connor and Strickland’s work [97]. Software defined radio are radios that can be controlled with software. Since many personal computers do not natively support reading or transmitting radio waves besides 802.11 and Bluetooth, adapters are required in order to be able to interface with those kinds of signals. Please note that prices may change, but they are accurate at time of writing.

- **RTL-SDR Dongle:** The RTL-SDR Dongle is a Sub-1 GHz low-cost SDR receiver used by O'Connor and Strickland [97]. As a low-cost option ($30 at time of writing [112]), this makes it more accessible even when class sizes are larger. It enables the reception of wideband radio frequencies, but it does not allow the transmission of any frequencies, so it is limited in that aspect.

- **Panda Wi-Fi 802.11 Dongle:** Panda Wireless [102] sells dongles to connect to Wi-Fi [97]. While many laptops may already contain some kind of network card, having a dongle like this can standardize the network capability of student machines. Since this device, like the RTL-SDR Dongle, is a relatively low-cost device with most of their devices being in the $30-50 range, it is more accessible to have many for a single class.

- **HackRF SDR:** The HackRF SDR is a half-duplex radio sold by Great Scott Gadgets [38, 97]. This gives the important advantage of being able to not only receive, but also synthesize and transmit radio signals. However, this radio typically costs around $300-400. Due to the significantly higher cost relative to the RTL-SDR Dongle, this is less suitable to be distributed en masse to students, and may be better suited as an in class activity only.
• **BladeRF SDR:** The BladeRF SDR is a full-duplex radio produced by Nuand LLC [96]. OConnor and Strickland used this device in order to set up a GSM (Global System for Mobile Communications) base station [97] in their classroom. The advantage of a full-duplex SDR over a half-duplex SDR is that it is capable of simultaneously transmitting and receiving. The cheapest BladeRF SDR costs $540 so it is a difficult product to use many of in a class.

4.3.2.2 Protocol Specific SDR

There are also a variety of SDR that are made to communicate using more niche or specific protocols. This can be useful when examining the security of internet of things (IoT) devices.

• **Attify APIMote:** Attify APIMote [110] is an example of a ZigBee capable SDR that is compatible with KillerBee [115], an open source framework for testing IEEE 802.15.4/ZigBee systems. The APIMote radio is developed by River Loop Security that also helps to develop and maintain KillerBee. While the cost of the APIMote at $149-190 is not as high as the HackRF SDR, at the time of writing they are currently out of stock.

• **YARD Stick One:** YARD Stick One (Yet Another Radio Dongle) is a sub 1-GHz radio dongle developed by Great Scott Gadgets [41]. According to OConnor and Strickland [97], the YARD Stick One is capable of working with KillerZee, which is a tool developed by Wright of the SANS institute, with its last release being in 2015 [147]. It costs around $100-150, depending on if you need to add an antennae or not.

• **Ubertooth One:** Ubertooth One [40, 39] is now retired product from Great Scott Gadgets, designed specifically to sniff Bluetooth frequencies and match the
Bluetooth frequency hopping. Because it is an open source project, it is possible to assemble the hardware manually, since it is no longer commercially sold. There are also alternative devices that can monitor Bluetooth and Bluetooth Low Energy (BLE), but Ubertooth One the most available documentation. At the time of writing, some Ubertooth Ones were available for resale at around $100.

- **nRF52840 Dongle:** The nRF52840 Dongle is a low-cost dongle based on the nRF52840 chipset developed by Nordic Semiconductors [86]. This is an alternative modern dongle available for sniffing Bluetooth frequencies, and it is typically sold for less than $15. It is one among several dongles for BLE, but serves as a possible alternative to the Ubertooth One. It is used by Nardi in this crash course tutorial on BLE sniffing [81].

- **CrazyRadio Dongle:** The CrazyRadio PA Dongle is a radio that can attack RF keyboard protocols developed by Bitcraze [6]. It is currently being replaced by a version 2.0, but the PA version is one of the three devices compatible with MouseJack [84], which is a collection of vulnerabilities for attacking RF mice and keyboards published in a whitepaper by Newlin of the Bastille Threat Research Team. This dongle is as affordable as RTL-SDR and Panda Wi-Fi 802.11 Dongle at $35.

- **USRP N210:** USRP N210 is a networked full-duplex SDR developed and sold by Ettus Research [28]. This was used by OConnor and Strickland with a VPN tunnel in order to make the classroom lab environment accessible to remote students [97]. However, this may not always be a feasible to set up as these radios are about $3,300 each.
4.3.2.3 Hak5 Hardware

Hak5 [42, 47] is a company that specializes in making penetration testing tools that make pentesting engagements easier. They are used commercially for offensive security engagements, so using them can provide realistic hands-on experience for future penetration testers. In some cases, it may be more educational and cost-effective to recreate the effects of their tools with a variety of other cheaper hardware and open-source software options. Notably, Hak5 does offer an education discount, but at the time of writing their education discount verification system was not working. In this subsection, I will explain some of their more popular products, but because of the large amount of devices they have, I will not be covering all of them.

- **Wi-Fi Pineapple Mark VII:** The Wi-Fi Pineapple is a rouge access point (AP) that provides a dashboard to set up Machine in the Middle (MitM) attacks [48]. Additionally, it provides a variety of other features that are useful, such as passive monitoring of nearby devices. When it is in use, it is accessible from a web browser, so no additional software is needed to use it. It costs $120, and the enterprise version is even more expensive.

- **USB Rubber Ducky:** The USB Rubber Ducky is a keystroke injection device. It is programmed with a proprietary language called DuckyScript, but it is a relatively simple language and there exists a large variety of payloads written in it available on GitHub [46]. So despite the fact that it is a proprietary language, gaining some familiarity with it could be useful. It costs $80, but there do exist cheaper alternatives if one is willing to put in the effort to recreating the device. One common alternative is using Digispark and flashing it with custom firmware [7].
• **Bash Bunny Mark II**: Bash Bunny is similar to the USB Rubber Ducky in that it is a USB attack vector for delivering a payload. However, the Bash Bunny is capable of imitating far more devices beyond just a Human Input Device (HID), for example Ethernet and storage [43]. This also allows for the exfiltration of data from a target after an attack. It starts at $120, which is similarly priced to other Hak5 devices.

• **LAN Turtle**: LAN Turtle SD is a remote access and network monitoring tool that is housed in a USB-A to Ethernet adapter [45]. It allows for many of the features of the Wi-Fi Pineapple, but in a wired configuration. It also markets itself as a tool that can be useful for both penetration testers and system admins, though it is unclear as to in which capacity it is more frequently utilized. It costs the $80, the same as the USB Rubber Ducky.

• **Key Croc**: Key Croc is a key logging device that can capture keystrokes and trigger payloads to exfiltrate data [44]. While this device requires physical installation in between the HID and the target, it could still be an interesting device to familiarize students with the attack vector of a keylogger. Like the Bash Bunny and Wi-Fi Pineapple, it costs $120.

### 4.3.2.4 Other

There are a couple of other hardware components that might be useful in conducting a security course. These are tools that are not SDR or belong to the Hak5 ecosystem of products, but have their own uses and purposes.

• **Raspberry Pi**: A Raspberry Pi is a general-use computer that can be useful for a variety of things, including hosting remote lab access if it is necessary for students to have access to a remote setup for hardware [67]. It costs $80, but
its flexibility means that it can be utilized for a variety of tasks, which makes it rather cost-efficient.

- **Flipper Zero**: The Flipper Zero is a portable multifunctional tool made for penetration testers and enthusiasts, encased in a toy-like form factor. It is used to read a wide variety of wireless signals such as RFID and radio, and replay them. It also contains some capabilities that enable hardware hacking. With a price of $170, it is on the pricier side, and while it contains many useful and interesting features consolidated into one device, the consumer-oriented nature of the device may make it less effective as an instructional aid for its price.

- **Pwnagatchi**: Pwnagatchi is an open source project that creates a reinforcement learning based AI model to crack Wi-Fi keys as a kind of pet [72, 73]. While it is part of the inspiration for the Flipper Zero, it is not itself a commercial product, and rather needs to be built from components that you can find on the website. The total cost of the components can vary depending on what quality components are purchased, but will likely cost a minimum of around $70-80. The AI component of the Pwnagatchi can also be disabled, and it can just be used as a device to learn how to attack Wi-Fi based devices. Additionally, if the course is more hardware engineering focused, there may be educational merit in learning to construct the Pwnagatchi as well.

- **HydraBus**: HydraBus is an open source hardware tool supported by HydraFW (Hydra Firmware) that is meant for learning how to hack embedded hardware [56]. It is extensible with a HydraNFC shield to allow it to interact with NFC (Near Field Communication) as well [57]. The device itself is around $50-60 dollars, depending on vendor, and the HydraNFC shield is an additional $110. Overall, it is a useful device for learning hardware hacking skills, but like the
Pwnagatchi the process of setting it up may be a bit more involved than some of the other products discussed in this section.

- **Proxmark**: Proxmark is a flexible RFID (Radio-Frequency Identification) tool which can be used for sniffing, cloning, and spoofing RFID signals [108]. It is a device that is designed more with professionals in mind, and the higher price tag of $340 reflects that. While it is fairly specialized and expensive, it can be a useful tool if RFID is a major component of the course. It is also open-source and has several extensible modules for interacting with other protocols as well.

### 4.3.3 Software

Once a platform(s) have been selected, there are a variety of software applications that can be selected. These can be broadly grouped based on their purpose, but also need to be evaluated on their availability and maintenance. This is by no means a comprehensive list of all software tools that can be utilized for ethical hacking, but I have tried to make it as diverse as possible to cover a wide variety of security topics.

#### 4.3.3.1 Traffic monitoring

One important step in ethical hacking is reconnaissance, and frequently this requires the monitoring of network traffic. There are many kinds of tools that enable this, which we will cover in this section.

- **Wireshark**: Wireshark is a widely utilized open-source network protocol analyzer licensed under the GNU General Public License compatible with major operating systems like Windows, Mac, and Linux distributions. It analyzes packet data in formats such as pcap and captures real-time data using the host
### Table 4.3: Software Reference

<table>
<thead>
<tr>
<th>Category</th>
<th>Software</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Traffic monitoring</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wireshark</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>airmon/airodump</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>Omnipeek</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>Debookee</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>dsniff</td>
<td>Individual</td>
<td></td>
</tr>
<tr>
<td>Kismet</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>Acunetix</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>Nmap</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td><strong>Password Crackers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aircrack</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>Reaver</td>
<td>Individual, Community Maintained</td>
<td></td>
</tr>
<tr>
<td>hashcat</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>John the Ripper</td>
<td>Open Source/Commercial</td>
<td></td>
</tr>
<tr>
<td>THC Hydra</td>
<td>Individual</td>
<td></td>
</tr>
<tr>
<td><strong>Rogue AP</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infernal Twin</td>
<td>Individual Owner</td>
<td></td>
</tr>
<tr>
<td>Wifiphisher</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>dsniff</td>
<td>Individual</td>
<td></td>
</tr>
<tr>
<td>Wi-Fi Pumpkin</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>Ettercap</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td><strong>Packet Manipulation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scapy</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>Hping/Nping</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>CommView</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td><strong>Binary Exploitation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>angr</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>Ghidra</td>
<td>Governmental</td>
<td></td>
</tr>
<tr>
<td>IDA Pro</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>Binary Ninja</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td><strong>Testing Software</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core Impact</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td>Metasploit</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>Burp Suite</td>
<td>Commercial</td>
<td></td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sectools</td>
<td>Individual</td>
<td></td>
</tr>
<tr>
<td>Social Engineer Toolkit</td>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>Universal Radio Hacker</td>
<td>Open Source</td>
<td></td>
</tr>
</tbody>
</table>
device’s hardware. The tool’s interface allows users to examine specific packets, apply filters, and conduct advanced operations, including filtering for specific IP addresses or protocols like TCP. It can be used as a command line tool, but it has a robust graphical user interface that is fairly easy to use as well. Wireshark natively supports decrypting protocols across 802.11 standards and extends its capabilities to non-802.11 standards like Ethernet and Bluetooth [146];

- **airmon/airodump**: Airmon and Airodump are part of the Aircrack-ng suite of tools, and together they allow you to monitor and record 802.11 traffic. Airmon is used to enable monitor mode on a device’s wireless interface, and you can use it to stop and start specific interfaces in monitoring mode. It can also be used to check if there are processes currently utilizing the interface [26]. Airodump is used to capture 802.11 frames and saves them in a format that is used with Aircrack. It writes to a text file which contains the details of the access points and clients that it observed while monitoring the traffic detected by a specific interface. Doing so requires that the interface being used be put in monitor mode. It is a fully command line interface, though while it is running, some controls can be used to change the output. It can also be run with a variety of options to change what packets to filter, write intervals, the bssids/channels to monitor, etc. [27].

- **Omnipeek**: Omnipex is a paid network analyzer tool available on Windows 7, 8, 10, and server. It is the self-proclaimed “World’s Most Powerful Network Protocol Analyzer” and allows for real time analysis of a wide range of network traffic types. It has a variety of graphical displays and other visualization tools, they also sell a USB WLAN (Wireless Local Area Network) device to aid in wireless packet capture. It offers some use cases for monitoring distributed networks as well [66].
• **Debookee**: Debookee is another paid network traffic analyzer, but it is available exclusively for macOS. There is a free trial that offers some of the features i.e. network scanning, intercepting traffic in the same subnet, and tcp port scanning, but obfuscates most of the results. Split into four modules: Network Analysis, Wi-Fi Monitoring, SSL/TLS Decryption, and “Professional” Module which is a bundle of the previous modules. Modules can be bought individually for a private or enterprise license [23]. There is pretty extensive documentation, and it seems pretty well-supported. It also utilizes Wireshark under the hood for some purposes, and can be used in conjunction with Wireshark to monitor more traffic.

• **dsniff**: dsniff is a collection of network auditing and pentesting tools. It was written by Dug Song, originally designed for his own purposes. It is built for Linux, though there exists unofficial ports for both Windows and macOS. The last official release was in the early 2000s. The dsniff module itself is used to monitor wireless traffic for passwords passed through, and there are other modules that can monitor wireless traffic for other kinds of interesting data. Additionally, it also contains several active pentesting modules that can perform spoofing and machine in the middle attacks [122].

• **Acunetix**: Acunetix is a paid website auto analysis tool that automatically scans a website for possible vulnerabilities. Acunetix is more of an application pentesting tool, and it is designed so that it can be built into the development process. It is not the most relevant to wireless pentesting, but it can be used by pentesters to run automated tests, and then manual testing can be used to follow up on possible vulnerabilities [58].

• **Nmap**: Nmap is a classic pentesting tool that has been in use for a long time, and is a free and open source project that is useful for network discovery and
auditing. It is able to gain more information about what devices are on a network and what services are running. It also has some other utilities that come with it, such as a graphical interface that allows you to visualize results. Not only is it widely used, well documented, well-supported, but also there are not many tools that can fulfill the same tasks as well as Nmap does [69]. Furthermore, It is also relatively widely used by the universities I surveyed in table 3.5

4.3.3.2 Password Crackers

Another common task in ethical hacking is the need to be able to figure out passwords. There are some tools that are specific to Wi-Fi keys, but also some general use password crackers that can be used against password hashes.

- **aircrack**: Aircrack is a commonly used tool that is in many courses used to recover WEP keys. It utilizes the PTW (Pyshkin, Tews, Weinmann) approach or the FMS/KoreK approach against WEP (refer to [64] for more technical details). WPA/WPA2 PSK cracking only works with dictionary attacks. Aircrack is a very commonly used tool that still is maintained, and it is used in many different tutorials [20]. It is a fairly easy to use tool, and works very easily with the rest of the aircrack suite [2].

- **Reaver**: Reaver is a now open source tool originally released in 2011 by Craig Heffner that conducts attacks against WPS (Wi-Fi Protected Setup) PINs. It conducts a specific attack against WPS because of a vulnerability in the PINs of WPS. Depending on the kind of WPS, either an online or offline attack can be conducted, though the offline one is significantly faster. It also includes a tool called wash to use a monitor mode interface to scan for networks [126].
• **hashcat:** Hashcat is a modern password cracker that utilizes both CPU and GPU. It supports a large amount of encryption algorithms, and it supports dictionary, brute-force, and rule-based attacks against hashes in the hc22000 format. It also claims to have the “world’s first and only in-kernel rule engine.” It is a more general purpose tool for attacking hashes in general, so it requires that captured packets be put into a format that the tool can process [50].

• **John the Ripper:** John the Ripper is a general use password cracker that is open source and has existed since 1996. It is a CPU based password cracker, but because of its long history, it has support for many hash and cipher types. It is Linux and macOS based, but it does have other versions created for other operating systems [98]. John the Ripper does offer a paid pro version of the software, but generally using it is free. The website also contains many word lists for password cracking.

• **THC Hydra:** THC Hydra does parallel brute force attempts to authenticate into a server and performs an online attack. It leverages multithreading and supports numerous protocols. It is still actively maintained on GitHub, but the documentation is only a single README text file and there are not many detailed tutorials on how to use it [139]. However, if you look at table 3.5, you will notice that both Georgia Tech and CMU use it in their courses.

### 4.3.3.3 Rogue AP

Rogue AP (Access Points) are tools that allow the attacker to get users to connect to it rather than a safe access point and conduct exploits on the connected users.

• **Infernal Twin:** Infernal Twin is a python2 rogue access point tool that has a GUI. It is used to eavesdrop the user’s traffic after they connect. It is very
broad in its usages, and also utilizes existing tools like SSLStrip. It was last updated in 2018, and while owned by one developer “entropy1337,” there are other contributors to the project as well. It is designed to help automate many of the common attacks rogue access points are capable of [71].

- **Wifiphisher:** Wifiphisher is another rogue access point framework also written in python used to perform phishing attacks against connected clients, and it was last updated in 2018. It has a couple of tutorials and has been presented at a couple of talks as well. The GitHub has a fairly comprehensive README, and it also has python docs on how to use it as well [12].

- **dsniff:** dsniff is covered in the traffic monitoring section, but it also contains some active spoofing and machine in the middle attack modules such as dnsspoof which forges replies to DNS queries on the LAN. However, It has not been updated since the early 2000s [122].

- **Wi-Fi Pumpkin:** Wi-Fi Pumpkin is a python3 tool for Linux based systems to conduct rogue access point attacks and machine in the middle attacks. There is some fairly extensive documentation and tutorials, and it has a discord server for developers. It has been updated in 2022, and is undergoing active development [101].

- **Ettercap:** Ettercap is an open source C based tool for man in the middle attacks. It is maintained by 3 primary developers, and it can conduct data injection and has some sniffing modules as well. There is documentation on the README of the source directory, but it is otherwise somewhat hard to find detailed information about it [100].
4.3.3.4 Packet Manipulation

The building blocks of network communication are packets, and there are several tools that can be used to craft packets for a variety of attacks.

- **Scapy**: Scapy is a flexible Python-based packet manipulation tool and library that allows users to craft, decode, send, and capture network packets. It enables the construction of customized packets for various network protocols and provides tools for analyzing and interacting with network traffic [114]. Scapy’s versatility makes it a valuable tool for creating and manipulating packets at different OSI layers. Scapy also claims to be able to replace many other tools such as: ”hping, 85% of nmap, arpspoof, arp-sk, arping, tcpdump, wireshark, p0f, etc.” Particularly if your university’s curriculum already includes python, Scapy can be a useful general use tool for interacting with packets.

- **Hping/Nping**: Hping is a command-line tool used for network testing and auditing developed by Sanfilippo, who is also the creator of Redis [113, 68]. It provides functionalities such as sending custom ICMP, UDP, and TCP packets, making it a versatile tool for network troubleshooting, security testing, and packet crafting. Hping allows users to specify various packet parameters, including source and destination addresses, packet size, and protocol-specific options. It is a relatively straightforward and lightweight command-line utility, but the wiki was last updated in 2009, and it is not currently maintained. Nping is a more modern alternative to Hping, developed by the Nmap Project [74]. Nping shares similarities with Hping, but comes with additional features and improvements. It supports IPv6 and has an echo mode that allows you to see how packets change in transit. It is also open-source as part of the nmap project, which means it is significantly more actively maintained.
• **CommView:** CommView is a commercial software tool for Windows that is used for monitoring network traffic [128], but one of its features is also a visual packet builder, which is mentioned by Trabelsi in several of his papers [132, 131, 130]. This can be particularly useful if students are likely to not have much familiarity with either python or command line tools. However, due to it being a commercial product, this would add to the operating costs of the class as it costs $300 per-user license without any add-ons. Trabelsi also mentions in his work an online FrameIP packet generator tool that fulfills a similar purpose [132, 131], but the website is in French and I could not find it.

### 4.3.3.5 Binary Exploitation

Binary exploitation is the process of manipulating vulnerabilities in executable binaries to gain unauthorized access or control. Doing so often requires some specialized software to analyze and exploit binaries.

• **GDB:** GNU Project Debugger (GDB) is a tool that allows you to see what another program is doing while it is running. While this is often a part of learning to do software development in C, it also has applications in hacking [33]. It is a free, open-source project, and if your university curriculum already includes learning C software development, students may already have some familiarity with it.

• **IDA Pro:** IDA Pro is a commercial reverse engineering tool sold by HexRays [54]. It is capable of decompiling a variety of processor architectures into assembly or close to source code. It also has a built-in debugger that you can use to step through the program while it is running. IDA Pro has been in development since 1990, so it well over 30 years old [55]. This lends it significant
credibility, however, the commercial nature of the software means that cost may make it expensive to use. They do offer a free version which supports a limited number of architectures, as well as an educational license that has capabilities somewhere between the free and pro version.

- **Ghidra**: Ghidra is an open-source reverse engineering tool developed and maintained by the National Security Agency (NSA) [82]. Similar to IDA Pro, it provides a set of tools for analyzing the functionality of binary executables, to decompile, analyze, and explore the inner workings of software applications. Notably, Ghidra is completely free and so it may be easier for students to use on their own time or post graduation, but some students may be hesitant about downloading software that is developed by the NSA.

- **Binary Ninja**: Binary Ninja is a commercial reverse engineering tool developed by Vector 35, with the very notable feature of having a free web-based client [140]. While the actual software requires a license, the online version is free to use, but it does mean submitting binaries to the developers of Binary Ninja. Downloading the demo version is also free, but it has limitations on analysis time and the architectures it supports.

- **angr**: The angr is a binary analysis framework developed primarily by the Computer Security Lab at UC Santa Barbara and the Laboratory of Security Engineering for Future Computing (SEFCOM) at Arizona State University [118]. It has capabilities for symbolic education and control-flow graph recovery, and it also has extensions to automate tasks like ROP chain building. It may have a bit of a high learning curve, but it can be very useful for exploiting binaries. It is important to consider how much learning students will gain from utilizing angr though, since it makes many binary exploitation techniques significantly easier.
4.3.3.6 Testing Software

There are also software tools designed around exploiting vulnerabilities, which provide a useful framework for ethical hacking.

- **Core Impact**: Core Impact is a paid tool that is used to automate many parts of penetration testing [30]. It costs around $10,000 a year to have a license to use, but it claims to be able to automate common and repetitive tasks. It has a library of exploits that it is capable of utilizing as well. It is not really designed for the individual and is geared more towards enterprise and corporate customers. As such, it is also difficult to find much information about what exactly its capabilities are.

- **Metasploit**: Metasploit is an open-source penetration testing framework developed by Rapid7 [109], and it has a plethora of available documentation, tutorials, and wikis. It uses a library of payloads in order to take advantage of vulnerabilities in software. It is under constant development and has many different modules and features and is free to use.

- **Burp Suite**: Burp Suite is a proprietary web application security testing toolkit, developed by Portswigger [107]. It gives you the ability to intercept incoming and outgoing HTTP requests, as well as the ability to modify those packets. It contains a variety of proxy and scanning tools in order to analyze web applications. While the commercial software is $450, they do have a community edition that is free to use, even if it does come with some limitations.
This last section includes tools that are useful, but do not fall neatly into any of the other categories.

- **sectools**: The site sectools.org is a website maintained by the developer of Nmap, Fyodor. It contains very brief summaries of different security tools with reviews from nmap users. It is not always the most up to date, but it usually links to documentation of tools that may sometimes be hard to find. It is mostly maintained by one person, though user reviews contribute to the site and can be useful as reference [70].

- **Social Engineer Toolkit**: The Social-Engineer Toolkit (SET) is an open-source penetration testing framework designed to facilitate social engineering attacks developed by David Kennedy of TrustedSec, SET automates and streamlines various social engineering techniques such as phishing, credential harvesting, and malicious USB drops. It helps security professionals test and strengthen the security awareness of individuals within an organization by emulating real-world attack scenarios. While significantly less technical than most of the tools mentioned, it is a significant part of penetration testing that has resources to support teaching it.

- **Universal Radio Hacker**: The Universal Radio Hacker (URH) is an open-source software tool designed for analyzing and decoding signals transmitted over radio frequencies, developed by Johannes Pohl [106]. URH provides a GUI to explore, demodulate, and decode a wide range of wireless communication protocols. It is compatible with many of the SDRs and Dongles covered in table 4.2, and may be necessary if they do not come with their own software.
4.4 Course Design Framework

Finally, by combining the ethical with the tables for platforms, software, and hardware, we can create a framework to help guide the creation of ethical hacking courses.

The framework I propose consists of six steps:

1. Identify course topics
2. Create relevant ethical/legal module(s)
3. Identify required platform(s)
   (a) Identify necessary hardware if any
4. Select software tools to be taught
5. Organize modules
6. Create lesson plans and schedule

4.4.1 Course Topics

The first step to developing a course is to determine what topics should be covered. Some courses can be developed to be broad intro classes that briefly cover a large variety of ethical hacking topics, while other courses can be developed to be more specific and cover a specific avenue of attack e.g. binary exploitation, web security, etc.

One resource to consider is the penetration testing KU provided by the NSA [88], as if achieving an NSA CAE-CD certification is a goal of your university, having a
course cover those topics will count towards certification. The 10 required topics are the following:

1. Flaw Hypothesis Methodology
2. Other methodologies (e.g., Open Source Security Testing Methodology Manual (OSSTMM))
3. Identifying flaws from documentation
4. Identifying flaws from source code analysis
5. Vulnerability Scanning
6. Understanding families of attacks
7. Understanding flaws that lead to vulnerabilities
8. Enumeration, foot printing
9. Attack Surface Discovery
10. Attack Vectors

Even if you are not attempting to achieve NSA CAE-CD certification, this list can be a useful jumping-off point to consider potential topics.

Beyond the NCAE-C designations, there are also ways to group topics for a specific course. For example, many of the universities surveyed had security classes specific to web, systems, networks, and hardware. While not listed in table 3.3, both Cal Poly Pomona and SLO offer a course on wireless security, and CMU’s Ethical Penetration Testing class covers Wi-Fi and IoT in one of its modules. Some additional topics that could be covered include social engineering, utilizing something like the social
engineer’s toolkit; traffic monitoring, whether passive or with a machine in the middle attack; or physical device security, teaching how devices like those vended by hak5 can be used.

4.4.2 Ethical Modules

Before teaching any technical skills, the first thing that should be covered is the ethical considerations. While there are many ways to go about this, I list several here to consider.

The simplest mitigation that can be implemented is creating a code of conduct for students to sign. This serves a dual purpose as it explicitly defines what kind of behavior is acceptable, while also serving as a paper trail to show that as an instructor you have done some due diligence in ensuring that students are informed on the possible consequences of their actions.

An additional mitigation that some classes do in Cal Poly is to require students to write an ethical statement when writing papers or report. Doing so causes students to consider how the research or work they are doing may have potentially damaging effects, and how to conduct security research while minimizing the chances for negative effects. An ethical statement is written at the beginning of a paper to disclose to the reader how the activities in the paper were conducted with ethical considerations in mind. It is particularly important to include such a statement when describing how to exploit a vulnerability.

An ethical statement should not only describe what mitigations were put into place in order to minimize potential harm to others while conducting research, but also what risks publishing a given paper may bring. For example, if a paper is being published that examines a new vulnerability, an ethical statement should explain
how the vulnerability was tested without causing any harm, how steps for responsible
disclosure of the vulnerability have been followed, and why it is not a significant
security risk to publish the paper at the time of writing.

As mentioned in section 4.2, there are also some ways to motivate students to act more
ethically. Summarizing some of the finding is Pike’s work, many security professionals
suggest that students are more inclined to act unethically when they are not given
chances to use their skills in an ethical manner. To this end, having an asynchronous
CTF challenge in the class can give students a safe way to practice the skills they have
learned and challenge themselves while not doing classwork. Additionally, connecting
students with student organizations in the university can also be a way for them to
receive positive social for utilizing and/or sharing their skills.

Finally, as an instructor, the lab infrastructure should be configured as not to interfere
with school infrastructure. Doing so requires careful consideration of the platforms
to use, which is covered in the following section.

4.4.3 Platforms

Once the topics are decided and ethical concerns considered, follow the flowchart in
figure 4.1 to identify what kind of platform you should utilize.

Refer also to table 4.3.1 for a guide to some of the different kinds of platforms available
to you to use.

4.4.4 Hardware

If you require the use of hardware, determine what that hardware is with the aid of
figure 4.2 and where to acquire and store it.
Figure 4.1: Flowchart to Help Determine what Platform(s) to Use

Figure 4.2: Flowchart to Help Determine what Hardware to Use
Refer also to section 4.3.2 if you want to review some of the different kinds of hardware available to you to use.

4.4.5 Software

Then, you can decide which software tools you want to teach during your course. Doing so may be a bit more complicated than picking platforms and hardware, but figure 4.3 may help guide the decision-making process, but there are a plethora of other software options to cover more niche use cases and may be better suited for your use case.

Refer also to section 4.3.3 if you want to review some of the different kinds of software available to you to use.
4.4.6 Organize Modules

The next step is to create modules to subdivide the course. This helps to keep topics manageable for students and can help you make sure you do not introduce too many tools at once.

4.4.7 Lesson Plans

Finally, adapt your modules to your specific needs as an instructor for the term being taught. This part will vary greatly depending on your institution and the schedule of the class. Determine how many class sessions you want to allot to each module in order to complete the entire course on time. You can allocate buffer time in case you end up spending more or less time on certain than expected. Other specifics that can affect your overall schedule include holidays, guest speakers, etc. Make sure to factor in configuration time when it comes to students setting up new tools and familiarizing themselves with new environments.

4.4.8 Summary

By following this framework, you can create a set of learning modules to achieve course objectives in covering selected topics. These modules will be supported with knowledge of what tools to utilize during each module and platform to base it all on. Finally, those modules can be converted to specific lesson plans and a teaching schedule.
In this chapter, I will discuss the results and note potential blind spots in my research methodology. First, we will talk about some of the difficulties with data collection and limitations present in the course development framework. Then, we will discuss other overall difficulties. Finally, we will summarize our findings and propose potential future work.

5.1 State of Ethical Hacking Education

Through my data collection for the state of ethical hacking education, I found that there has been a trend of increasing interest in funding from the NSF towards advancing the field of cybersecurity, and there have also been more and more universities being certified through the NSA’s CAE program. Both of these suggest that there is an interest in the educational field to teach more cybersecurity. However, it is prudent to mention that neither of these are clear indicators of specific focus on the topic of ethical hacking. Additionally, there are many additional sources of funding besides NSF grants that could also indicate more or less focus on cybersecurity, but in order to limit the scope of this thesis since it is not a comprehensive analysis of educational funding, I only examined the NSF. Similarly, not being accredited by the NSA’s NCAE-C program is not necessarily an indicator of a lack of security education. Despite these blind spots, this does not invalidate my findings because any additional indicators of security education funding or certifications can only have a positive effect on the total amount of institutions demonstrating interest in teaching
cybersecurity. Therefore, my results serve only as a conservative minimum, yet they are still indicative of a growing field of cybersecurity education.

5.2 Course Development Framework

While I hope that the course development framework can serve as a useful resource for developing ethical hacking courses, it is important to acknowledge some of its limitations. First is also the problem of scope. Because the possible ways to design a course are numerous, and the resources available for ethical hacking are too, it is infeasible to design any course that covers every aspect of ethical hacking. Rather, different flavors of courses need to be designed to focus on different topics within the domain of ethical hacking. The framework is also by no means a comprehensive list of all the possible platforms, hardware, and software that could be used to teach ethical hacking, as new tools are constantly under development, so a comprehensive list is unattainable.

Additionally, by itself, utilizing the framework will not be sufficient to finish designing a course. By design, it primarily focuses on lab and project design, not course content. I also do not provide any strong recommendations for any tools mentioned in the framework, leaving the judgement of what tools to use to the instructor. This allows it to be broadly utilized as a reference by a variety of instructors who may have access to different resources.

The framework is also limited due to its lack of verification. I was unable to do so due to limitations in scope and time, but potential future work could include conducting case studies in order to determine the validity and usefulness of this framework in designing courses.
5.3 Other Difficulties

Another difficulty I encountered when writing this thesis is its interdisciplinary nature. This thesis incorporates topics including computer science, ethics, legality, and pedagogy. My primary area of study is computer science, so that is where I have the most subject matter expertise. From an ethical perspective, there is room for the framework to be expanded upon with a systematic review of the ethical considerations of hacking using ethical first principles. There is also additional room for a more in depth look at potential legal liabilities and how that might change state by state depending on local laws. Finally, while I have learned about computing pedagogy in theory, the best way to evaluate the framework would be to design and carry out a course using it as a reference, which I did not have the opportunity to do.

5.4 Conclusion

In this thesis, I conducted a survey of funding and certification data, as well as an investigation into 14 universities demonstrating a trend of investment into cybersecurity education. Then, I presented a course development framework that provides a way to scaffold the process of developing a course for teaching skills relevant to ethical hacking. Within the framework, I provided a review of ethical and legal concerns that should be considered when teaching ethical hacking. I also provided a set of tables of the available platforms, hardware, and software that can be used for course infrastructure and content. Together, the contributions of this thesis serve as a useful insight into the current state of cybersecurity education and a reference for developing future courses in ethical hacking.
5.5 Future Work

There are a number of ways the work done in this thesis can be extended.

First, there is room for more thorough investigation into understanding the landscape of ethical hacking education. Since much of the available data is generalized to the field of cybersecurity as a whole, collecting data on specifically ethical hacking and penetration testing will reveal more specific trends. Examining the currently utilized tools by universities could also be expanded by reaching out to universities and their instructors in order to collect more data.

An investigation could also be done into non-governmental funding or the job market of penetration testing to improve understanding of what the market trends of ethical hacking jobs in the private sector are. The NSF and the NSA only demonstrates governmental interest into the furthering of cybersecurity education, and does not take into consideration private donors to universities or corporate sponsored research.

On the ethical side of things, with more understanding of different philosophical schools of thought, a more comprehensive and concrete way of developing ethical modules could be proposed. One could also try to develop a set of ethical lessons that can be included in a course, depending on course content. Currently, I do not have a good way of determining what ethical mitigations should be utilized in a course, and it is left to instructors to determine for their selves.

Legally, further work could be conducted by examining specific laws rather than only legal principles. Doing so could help provide useful legal advice for instructors when setting up and conducting courses.
Finally, utilizing the framework to develop new courses and carrying out those courses will help to validate and identify other improvements for the framework.


    original-date: 2021-07-13T22:03:56Z.


[78] MIT. MIT OpenCourseWare, 2023.


https://www.nsf.gov/awardsearch/advancedSearchResult?ProgEleCode=8060&amp;BooleanElement=Any&amp;BooleanRef=Any&amp;ActiveAwards=true#results.


[98] Openwall. GitHub - openwall/john: John the Ripper jumbo - advanced offline password cracker, which supports hundreds of hash and cipher types, and runs on many operating systems, CPUs, GPUs, and even some FPGAs, 2023. 43.


[120] SJSU. Catalog Search - San José State University - Acalog ACMS™, 2023.


Appendix A

COURSE MATERIALS

Below are all the references to university courses I used in my survey. First is all the course catalogs I referenced, and then next is individual course pages I examined if they were available.

- Course catalogs [135, 136, 137, 10, 9, 8, 120, 34, 138, 15, 78, 11, 123, 17]
- UC Berkeley [32]
- UC Los Angeles [104]
- UC San Diego [80]
- Cal Poly SLO
- Cal Poly Pomona
- San Jose State [119]
- Georgia Tech [145, 144, 127]
- UIUC [1]
- Arizona State
- Carnegie Mellon [60]
- MIT [111, 85, 53]
• Caltech

• Stanford [19]

• Cornell [141, 31]