Jared Robert Andraszek

Computer Science and Cyber Security

Professor Julie Henderson

May 2025

Problem Statement:

In this modern day of education, with technology rapidly changing, many students feel their college curriculum needs to keep up with what they want and need to know for computer science and cybersecurity jobs. While these degree programs explain the fundamentals and give an excellent baseline for students to develop further, we have few specialized skills that students can develop in the courses. Courses such as Network Security, Cyber Defense, and Principles of Cyber Security teach the fundamentals. Still, my experience with the hands-on lessons is that they treat the user as a machine that knows nothing and tells you what to do and how to do it, leading to a mind-numbing copy-and-paste experience. One example would be NETLabs, which follows a step-by-step methodology. While the lessons cover the requirements and the course's explanation, it assumes the student knows nothing. Thus, they act as if they know nothing, lowering user retention.

Some options that kept my attention and interested me in Cybersecurity with specific skill sets were competitions such as CyberSEED, SECCDC, and PCDC, as well as additional side lessons hosted by third-party educators like HackTheBox. These competitions and programs are open to users who know almost nothing but have more difficulties available for those with more experience. Capture the Flag (CTF) competitions like CyberSEED have hidden passphrases, or flags, that require users to break websites and find vulnerabilities to earn points and complete the challenges. Blue team competitions such as SECCDC and PCDC have pre-set devices, and a red team (professional attackers) is trying to break in. Finally, HackTheBox is an educational program that teaches vulnerabilities and concepts but tests them using a CTF related to the material. In real-world scenarios, these competitions that use problem-solving, trial and error, and teamwork are what students should learn.

The solution is a mix of HackTheBox and a CTF but with a story with varying difficulty levels that teaches a variety of Cybersecurity vulnerabilities and other concepts that assume the student knows the fundamentals and grows from there. There should also be multiple endings with options for computer ethics to come into play. This program would teach students in a fun and engaging method that would cover some of the more requested topics that students expect to learn from ethical hacking or penetration testing classes. This program would also be accessible to students for free and give the users the knowledge and skills regarding how specific vulnerabilities work and why they work.

Project Description:

This project will consist of five to ten virtual machines that host websites with vulnerabilities that players can attack and find the flags and mission orders. The idea for the game is to have a player connect to a Kali Linux virtual machine (VM) or have a VM with all of the mission files and necessary software. From that VM, they access the websites and complete the operation. All mission orders will be available on the VM's desktop. However, they will be encrypted using the flag from the previous mission. Each mission will have a specific vulnerability they will teach the user; for example, the first website will have a file traversal and access exploit that users can exploit by seeing the website's URL updates with the variable names and values or HTML GET form submissions. Along with the vulnerable website, there will also be a detailed write-up of the exploit, why it works, and how to prevent it. The user will progress to the next mission after successfully finding the key.

Each VM will host a website stored on CSU's ECXi server and accessible via the school's Cybersecurity lab. Each VM will have a unique IP address to add realism and a story narrative about why the user is hacking into the website. At the final mission, clues and context will be available to the player, bringing ethical debates and questions about their actions. Eventually, this leads to the user having a choice on which side they choose to win with: the morally questionable team the player starts with but is their job or the group they have been actively attacking. Eventually, they will make a decision and get a certificate of completion unique to the team they chose that they can print.

An instructor's manual will also explain the answers and how the students should make progress. This information is available for professors or instructors who want to use the program. Hints are also available to the players in the form of commands following the naming format "Mission#-Hint#." These hints have no change or effect on the game but do allow the user to progress without answering the problem for the user.

There will be, at a minimum, five websites and missions for players. There will be GET and POST exploits for file traversal, SQL injections on an unsecured website, buffer overflows to execute code, Wireshark for PCAP analysis, and NMAP for open ports and reconnaissance. There will also be a final mission that incorporates all of the previous lessons. However, there will be two websites for the final mission, depending on the ending the user is going for. They will be exact mirrors with appearance changes.

Proposed Implementation Language(s):

HTML, CSS, Javascript, SQL, PHP, and BASH

Libraries, Packages, Development Kits, etc:

Apache webservers

Additional Software:

Oracle Virtalbox, Nmap, and Wireshark.

Personal Motivation:

Over my junior year summer, I attended a cybersecurity internship called the Advanced Course in Engineering – Cyber (ACE-Cyber) program and there we learned by trial and error, making mistakes, and teamwork. While there I realized that it retained my attention and made learning fun. A good portion of ideas such as buffer overflows, SQL injections, and Nmap came from their curriculum. So, a group of friends and I came together to make a program that would allow users to do something similar to what this project is.

Credit for the idea goes to Aaliya Jakir from Georgia Tech. She came up with the idea to create vulnerable websites to attack and the initial concepts of this project. However, with school starting up and difficult meeting times, the group fell apart and all progress for the project stopped. I wanted to continue working on it, and while it doesn't meet the initial vision of the project, I wanted to make something that I could give to my friends and let them have fun while also learning why these complicated concepts work.

Schedule:

- Draft of Proposal	2/21/2024

- Revision of Proposal 2/28/2024
- Finalization of Proposal 3/13/2024
- Draft of Requirements 3/27/2024
- Revision of Requirements 4/3/2024
- Finalization of Requirements 4/23/2024
- Have story complete 5/1/2024
- Start working on the first website 5/1/2024
- Finish the first website 8/15/2024
- Finish the documentation of the website 8/16/2024
- Finish the second website 8/24/2024
- Finish the third website 9/10/2024
- Finish the fourth website 9/22/2024
- Finish the fifth website 10/2/2024
- Upload VMs to the ECXi server 10/10/2024
- Configure server and networking 10/24/2024
- Finalize User VM 11/10/2024
- Get user feedback 12/1/2024
- Implement changes 3/14/2025
- Make presentation 4/30/2025
- Presentation 5/1/2025