Creating the Pwnable Claw Machine

A Major Qualifying Project (MQP) Report
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements
for the Degree of Bachelor of Science in

Computer Science

By:

Arthur Ames
Charlotte Clark
Evelyn Dube
Declan Murphy

Project Advisor:

Professor Robert Walls

Date: March 2023

Abstract

Capture-the-flag competitions are a popular educational tool for cybersecurity. We designed and implemented a beginner-friendly capture-the-flag (CTF) to encourage participation in the cybersecurity program at Worcester Polytechnic Institute. The competition centers around a physical claw machine, where successfully solving challenges earns you a chance to use the claw and win a prize. We designed and built this claw machine, as well as integrated it with an existing open-source CTF website. We created a set of storylines to draw engagement with our CTF competition, and applied them to a series of existing challenges. Lastly, we adapted the CTF website to better fit our needs, incorporating new colors, theming, and functionality to create an exciting competition platform for students of all experience levels.

Acknowledgements

Thank you to our teammate Avery Smith for his contributions to the Pwnable Claw Machine.

Contents

1 Introduction **1**

2 Background **3**

2.1 Capture the Flag Structure 3

2.2 Existing Infrastructure 4

2.3 Software Security Engineering 7

3 Overview **8**

4 Physical Claw Machine **9**

4.1 Cabinet Design 9

4.2 Control Scheme 11

4.3 Claw Mechanism 11

4.3.1 Gantry System 11

4.4 Claw Electronics 14

4.4.1 High Level Control 14

4.4.2 Low Level Control 15

4.4.2.1 ESP32 Microcontroller 15

4.4.2.2 Stepper Motors and Drivers 17

4.4.2.3 Claw Electronics 19

4.4.2.4 Power Supply 20

4.5 Embedded Software 21

5 Design and Implementation of the Front End **22**

5.1 Expanding upon PicoCTF 22

5.1.1 Claw Machine Integration 23

5.1.2 Addition of Administrative Features 23

5.2 User Experience Design 24

5.2.1 Story Tie-Ins 26

5.3 Azure Integration 28

6 Story and Lore **30**

6.1 The Lore 31

6.2 Story Incorporation 34

6.3 Applying the Story 35

7 Future Work **39**

7.1 Pwnability of the Claw Machine 39

7.2 Audience and Content 39

7.2.1 Audience 40

7.2.2 CTF Content 41

7.3 Integrated Control Board 42

8 Conclusion **44**

References **45**

List of Figures

1 Before and after PicoCTF redesign 7
2 Reference image for claw machine design [13] 10
3 SOLIDWORKS rendering of the cabinet design 11
4 Spars on the cabinet for stability 12
5 LED lighting 12
6 Gantry System 13
7 Simplified differential belt drive diagram 13
8 Belt Tensioner Bracket 14
9 Motor Mount 15
10 Low-level hardware wiring diagram 16
11 Soldered ESP-32 with input/output labels 17
12 NEMA-17 Stepper Motors and TMC2209 Driver 18
13 The annotated protoboard motor drivers 19
14 The claw used in the machine 19
15 H-Bridge circuit diagram 20
16 ATX Power Supply 21
17 Moodboard created to refine the desired design aesthetic 25
18 Guide to the colors and fonts used in the redesigned site 26
19 Before and after site redesign 27
20 Four cute cats 28
21 Agency Report to introduce the story to students 36
22 Challenge description for stack0-64 36
23 Prototype PCB Design 43

1 Introduction

Capture the flag competitions (CTFs) are a popular tool in the cybersecurity education community. Combining practical techniques with teamwork and competitive spirit, CTFs are a structured, safe way to gain experience hands-on security techniques in a controlled environment. CTF creators can design problems to guide students through different techniques in a way that is engaging, without breaking the law.

WPI has a small but strong cybersecurity community. Courses in cybersecurity are primarily targeted to seniors due to the amount of background knowledge needed to succeed. Because of this, many underclassmen do not consider cybersecurity as a potential area of research, unless they arrived at WPI with a preexisting interest in systems programming or other security topics. Our goal is to make cybersecurity more appealing and accessible to students without prior experience in the subject. By exposing students to cybersecurity earlier in the academic career, they may find themselves interested in taking advanced coursework in the subject as upperclassmen. To achieve this, we are working to create an eye-catching, beginner-friendly CTF competition to increase awareness of the cybersecurity programs at WPI. We want to curate an opportunity to explore cybersecurity with a low barrier to entry, and hope to spark genuine interest from students who may be intimidated by the reputation of systems programming and cybersecurity.

Our design centers around a physical claw machine to be displayed in Fuller Laboratories. However, rather than quarters, this claw machine only accepts payment in the form of CTF flags. Students visit the associated website, solve challenges, and earn points which can be exchanged for attempts at the claw machine. We have also integrated this CTF into the platform used for CS 4401: Software Security Engineering, and much of our development was done with this class in mind. In this report, we give a high level overview of the various components of our claw machine and CTF system in Section 3. Then, we discuss

the development of the physical claw machine in Section 4, the function and aesthetic of the CTF website in Section 5, the story and lore of the CTF challenges in Section 6, and our ideas for future work in Section 7.

2 Background

2.1 Capture the Flag Structure

Cybersecurity Capture the Flag competitions typically fall into two categories: jeopardy-style and attack defense [17]. However, both types of CTFs usually feature teams competing for points, where the team with the most points at the conclusion of the event wins.

Jeopardy style CTFs are centered around independent challenges. The puzzles are created ahead of time, and may have different difficulty levels and point values. Challenges are often separated into categories based on the subset of cybersecurity they contain. In this scenario, teams earn points by defeating the challenges created by the CTF developers. It is purely a race to earn as many points as possible. This is in contrast to attack-defense CTFs, where teams compete head-to-head. One team has to defend a resource, while the other is trying to gain access. Defending points are earned by successfully patching vulnerabilities, and attacking points are earned by successfully retrieving data.

For our CTF, we will be adopting the jeopardy style of competition. Because our participants will have highly variable skill levels, having teams compete directly would not be an enjoyable experience. Additionally, our CTF is open ended rather than being bound to a fixed time period. Since WPI students are often busy with coursework, having challenges available around the clock will encourage students to participate on their own time, as blocking off an entire weekend for a competition is often unfeasible. Completing challenges independently enables participants to compete asynchronously, increasing the accessibility of our CTF.

2.2 Existing Infrastructure

For this project, we expanded upon an existing CTF infrastructure called PicoCTF. PicoCTF is a free cyber-security education program run at Carnegie Mellon University. It utilizes the jeopardy-style CTF format to expose students to challenges curated by security and privacy experts [26]. In addition to being free to participate, in 2018 they released the source code of their web-based competition software on GitHub [27]. While the existing implementation supports many features, such as inter-institutional competition, teams, and classrooms, for usage at WPI we wanted to both simplify and extend functionality of the PicoCTF framework. Before discussing the modifications we made to the codebase, first we need to explore the design of the original site. PicoCTF utilizes several different technologies throughout the web interface. At the highest level, Jekyll is used to compile a collection of scripts and HTML files into layouts specified by templates. The majority of these scripts are generated by compiling React into vanilla JavaScript, which contains both logical elements and the UI code. These React files utilize the ReactBootstrap library for easy UI components with consistent styling. This combination of technologies has both benefits and unique quirks that we adapted to throughout the development process.

Jekyll

Jekyll is a Ruby Gem that is used for the development of static sites. Its templating system makes it ideal to use for sites like blogs, where many pages share a common format. The key components of a Jekyll site are a _layouts folder containing templates, HTML files for page content, and front matter at the start of each HTML file which specifies the page title, which template to use, and other variables.

Example of Jekyll HTML Syntax

```
---
layout: default
title: Homepage
foo: bar
scripts:
    - /js/script.js
---
<div class="page-content">
    <h1> Hello, world! </h1>
    <p>
        This is an example of an HTML file using Jekyll.
    </p>
</div>
```

The use of layouts allows developers to avoid rewriting boilerplate sections which are needed across several pages of their site. In PicoCTF, the main use of the templating system is to share the same metadata and navigation bar across the entire site. The metadata attached to each page includes the language, character encoding, icons, and scripts. While many of the scripts are shared across all pages, the layout also allows for pages to include additional scripts, such as files containing the React components used within the page. These scripts are included in the front matter of the specific HTML file.

React

React is a JavaScript library for building UI components. Developers can create customized, modular UI elements which can then be reused throughout the site. According to Stack-

Overflow's annual developer survey, React is the most popular front-end web library [22], far ahead of comparable technologies such as Vue and Angular.

React's component-based system is useful when designing interactive applications. It uses a unique JSX syntax, which extends existing JavaScript syntax to include HTML-like code [20]. This makes it easier to associate logic with the rendered output. These JSX files are then converted into vanilla JavaScript, which can be run in the client's browser.

PicoCTF uses React for all UI elements throughout the site. However, it still uses Jekyll as the foundation. This is unusual, because typically React is used for the entire site [20]. PicoCTF integrates the React components with Jekyll templating by converting the JSX into JavaScript first, and then including the resulting scripts in the Jekyll front matter as needed.

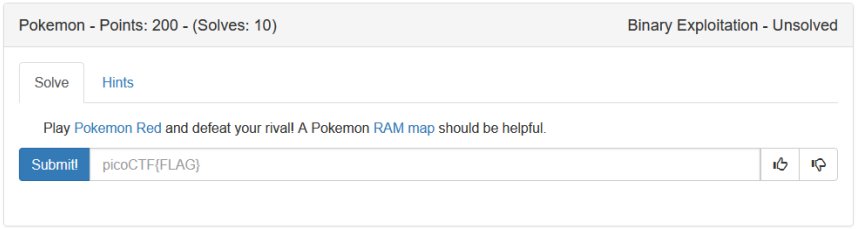
Other Dependencies

Other libraries are also used throughout the code, but are less critical to the overall structure of the site. For instance, Bootstrap components are used for most of the UI, and jQuery is used to map React components with where they are placed in the HTML. However, it is important to note how libraries must be included in the site.

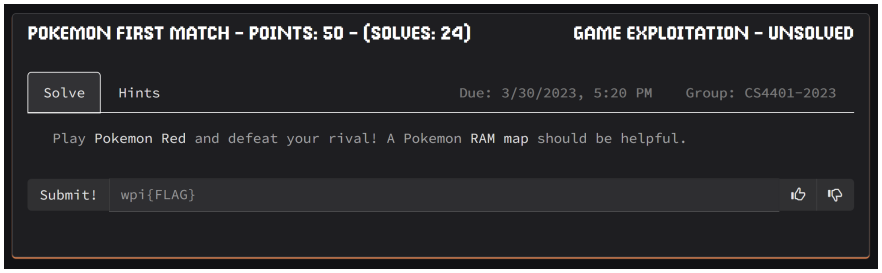
Because we are dealing purely with the front end, all of our scripts must run directly in the browser. This means that we cannot require third-party packages. However, packages are essential to developing for the web. To work around this issue, we download the JavaScript source code for the libraries we need, and add it into the project in the `/picoCTF-web/web/js/libs` directory. Then, we can include those files as scripts in the site's HTML. In our context, this means incorporating it into the Jekyll templates or front matter.

2.3 Software Security Engineering

As part of this project, we created and integrated storylines for challenges and a few other assignments from CS 4401: Software Security Engineering. Software Security Engineering teaches software and systems security via a CTF-style challenge system. Students learn about vulnerabilities in class, and then complete challenges that require them to detect and exploit those vulnerabilities. By creating storylines for challenges used for this class, we could informally gauge people's reactions to determine if the lore was engaging and entertaining. Software Security Engineering previously used the PicoCTF platform described earlier. It served its purpose, but could be improved both functionally and aesthetically. This project added many quality of life features that improved the experience for students as well as administrators. This project also made significant changes to the looks of the website to tie in with the arcade-style theme we desired.



(a) Original PicoCTF



(b) Our version of PicoCTF

Figure 1: Before and after PicoCTF redesign

3 Overview

This section briefly describes each major component of the Pwnable Claw Machine system, as well as their purpose. First, we have the physical claw machine. The claw machine provides a chance for prizes to users who complete challenges in the CTF. It displays scoreboards and announcements for the CTF on the monitor in the back. It connects to some of the storylines of our CTF via vinyl art of the characters on the sides and back. Lastly, it provides advertising for our CTF by being flashy, colorful, and interesting.

The front-end website presents the majority of the CTF content. It serves challenges to CTF participants, and is where students can submit flags for prizes and attempts at the claw machine. The website is where students would spend their claw machine attempts in order to start up the physical machine. The website is also the primary way in which players will engage with the lore, via challenges and the overall aesthetics of the site. The front end also allows for administration of the CTF via admin pages, which control which challenges are visible, what groups can access them, and more.

The back-end hosts the content required for both the physical claw machine and front end. It hosts the required files and servers for challenges, and handles the translation between claw machine attempts and starting the physical machine. Few significant modifications were made to the back-end during this project, so it is not extensively discussed in this report.

4 Physical Claw Machine

As previously stated, our goal is to make cybersecurity engaging and appealing to students who otherwise might not be exposed to the topic. Part of this includes the creation of a claw machine to dispense prizes for completing challenges. Our idea was that a large, colorful, flashy claw machine would draw attention to itself and get students interested. Physical prizes for completing challenges would also hopefully draw students in. Installing a monitor in the machine allows us to include more flashy graphics to draw attention, as well as broadcast scoreboards to include a competitive aspect. Our hope is that the claw machine would be a unique aspect of the CTF that would also help to draw in new players.

4.1 Cabinet Design

The exterior design of the claw machine was inspired by the style of Japanese arcade claw machines (seen in Figure 2).

By including large acrylic windows on the machine, we were afforded much more space to include graphics within the machine. This extra space also allowed us to include a large monitor in the back wall of the machine, which is used to display scoreboard and challenge information.

Since the cabinet needs to be semi-portable, we settled on the design seen in Figure 3. The cabinet is approximately three feet wide, and 6 feet tall. These dimensions allow an average height person to see the monitor at eye level. The width being three feet allows for stability and sturdiness. To ensure stability of the acrylic panels, we decided to attach spars to the front of the cabinet as seen in Figure 4. The spars are small enough to not be distracting, but still sturdy enough for the acrylic panels to be attached to. As an extra precaution several plywood pieces are attached with wood glue in addition to screws. For example, there is another set of internal spars in the bottom of the cabinet. These spars

Figure 2: Reference image for claw machine design [13]

hold the foundation of the cabinet together, and to maximize stability we used both screws and wood glue. The monitor is stabilized by 3d printed brackets on the front four corners and metal straps on the back. The stability of the gantry system, attached to the top piece, was not really a concern since any downward force from the weight of the electronics/claw is easily absorbed through the spars.

The electronics are housed inside the bottom portion of the cabinet. The cables, micro-controllers, etc. are accessed through the backdoor, which swings on hinges. Prizes can be added by un-mounting the monitor from the back face. The outside is painted black to contrast against the brighter colors of the vinyl art. The interior, on the other hand, is painted light grey so that the inside is easily visible and well lit. Interior lighting comes from both the monitor as well as LED strips as seen in Figure 5. The LEDs were routed around the inside of the cabinet with the LEDs facing the monitor where possible. This ensures the lighting is not distracting to a user, but also effective.

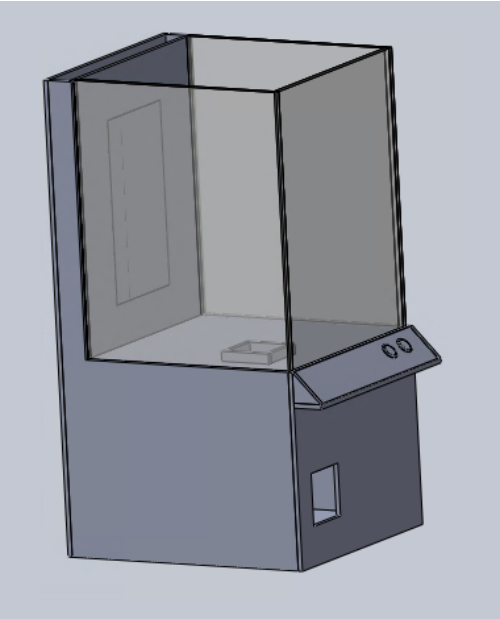


Figure 3: SOLIDWORKS rendering of the cabinet design

4.2 Control Scheme

There are two buttons on the cabinet for operating the claw. One button moves the gantry system left and right, and the second button moves the system forwards and backwards. Upon release of the second button, the claw immediately descends and attempts to grab a prize. This two button system aligns with the Japanese style of claw machine. American machines, on the other hand, tend to use a joystick to move the claw arbitrarily.

4.3 Claw Mechanism

4.3.1 Gantry System

To move the claw, a gantry type system was designed. The system consists of a small platform suspended from linear bearings mounted on two 800mm linear rods, which themselves are also mounted perpendicularly on another pair of 800mm linear rods. This allows the platform which the claw is mounted on to move anywhere within the 800mm x 800mm area of the

Figure 4: Spars on the cabinet for stability

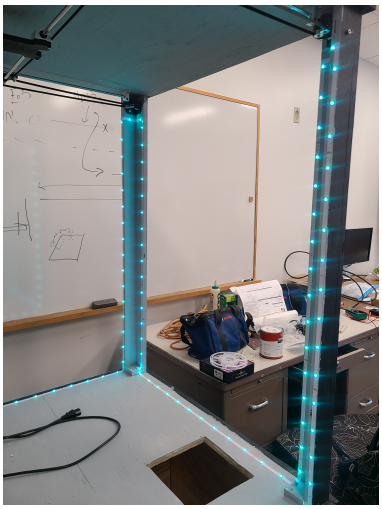
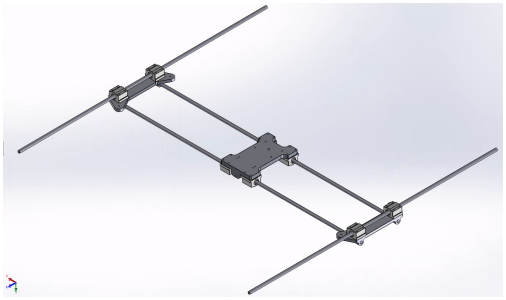


Figure 5: LED lighting

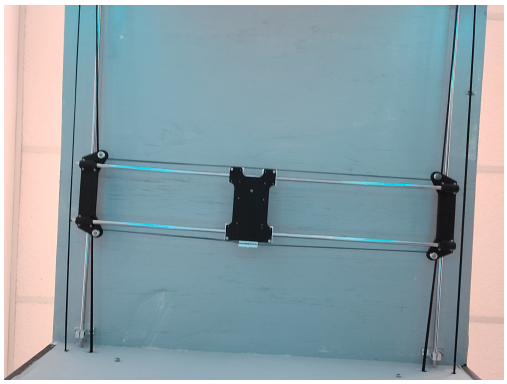
top of the interior of the machine.

The gantry is driven by two motors in a differential drive configuration. Both motors are mounted in the back of the claw machine, and belts are fed through holes in the back pane of the machine. By using a differential drive configuration, the motors can remain stationary in the machine, which both simplifies wiring of the system, and eliminates the inertia of the heavy motors.

Movement in both directions can be controlled by running the motors at different speeds and directions, as illustrated in Figure 7. When the motors are both run at the same

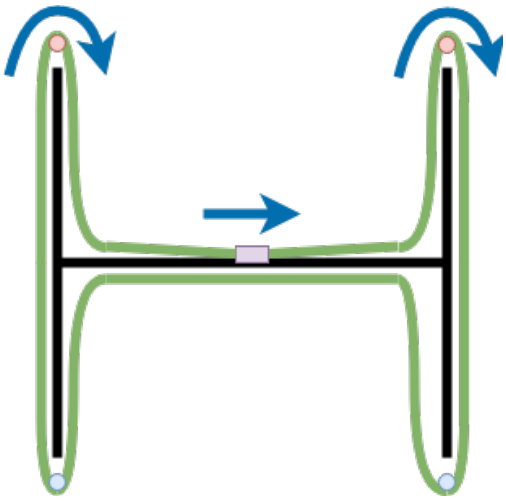


(a) SOLIDWORKS Rendering

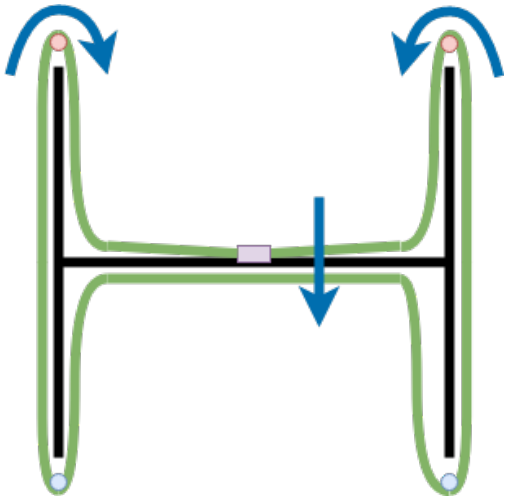


(b) Installed In Machine

Figure 6: Gantry System



(a) Lateral motion

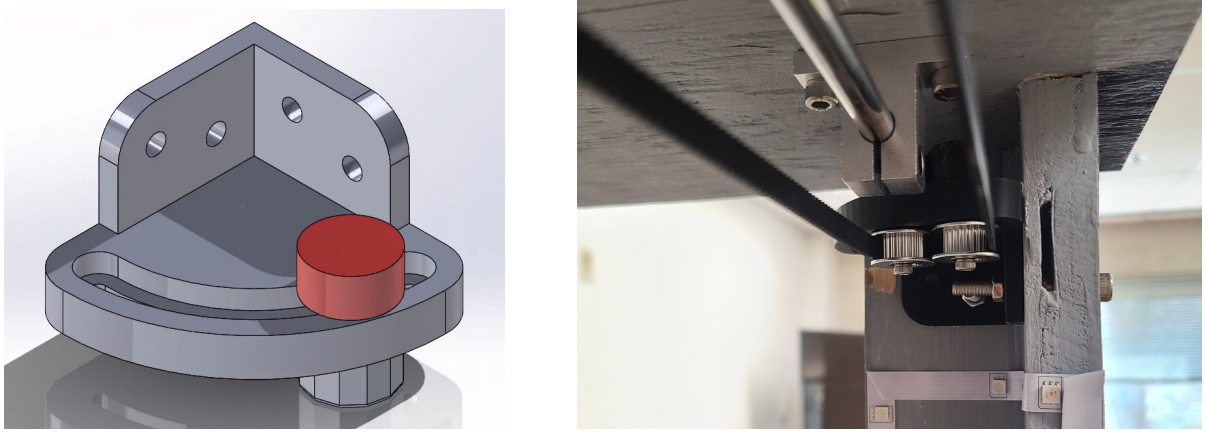


(b) Forward/Backward motion

Figure 7: Simplified differential belt drive diagram

speed in the same direction, purely lateral motion can be achieved. When motors are run at the same speed, in opposite directions, forward-backward motion can be achieved. By mixing speeds and directions, these motions can be mixed, allowing for any type of motion within the 2d plane of the gantry.

A 5 meter GT2 timing belt was used as the main drive belt, and custom 3d printed tension and motor mounts were created to keep the belt in place. All 3d printed components were printed using black PLA+ filament, at 20% rectilinear infill. Grooves were place in the parts with adjustable knobs, which then attached to off-the-shelf GT2 idler bearings. This allowed the position of the idlers to be adjusted easily after installation, so the belt could be properly tensioned.



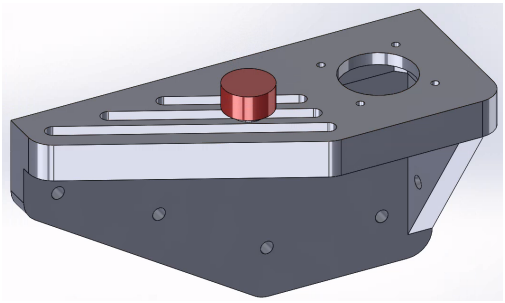
(a) SOLIDWORKS rendering (b) 3-d Printed Part

Figure 8: Belt Tensioner Bracket

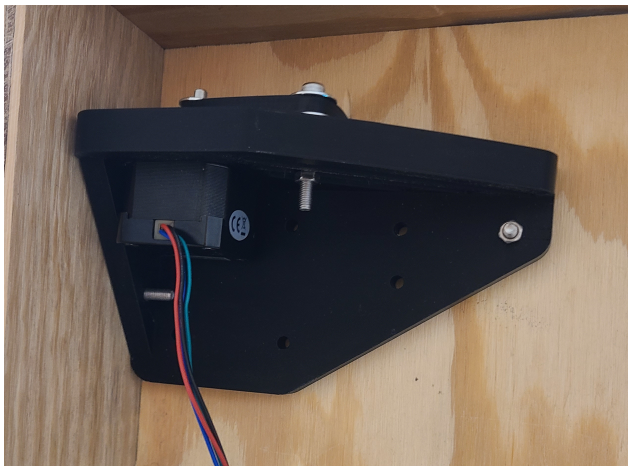
4.4 Claw Electronics

4.4.1 High Level Control

The "high level control" section of the claw electronics consists of a Raspberry Pi handling much of the communication between the CTF infrastructure and the claw machine.



(a) SOLIDWORKS rendering



(b) 3-d Printed Part

Figure 9: Motor Mount

When a user uses one of their claw machine attempts, the back-end tells the Pi to enable the claw machine. The Pi is also the translator between the signals from the button control panel and the ESP32 Microcontroller. Lastly, the Pi also controls the monitor to display scoreboards and other information.

4.4.2 Low Level Control

The “low level control” section of the claw electronics consist of the micro-controller, motor drivers, and other hardware responsible for moving the gantry and actuating the claw. The primary method of communication between the ”low level” and ”high level” sections of the electronics is a serial connection. When the high level electronics determine that the user is eligible for a prize, a serial signal is sent to the low level electronics that triggers the software on the micro controller to enable the claw.

4.4.2.1 ESP32 Microcontroller

Low level control of the hardware required that we include a micro-controller that was able to run firmware to coordinate the motors, claw, and other electronics. For this purpose, we

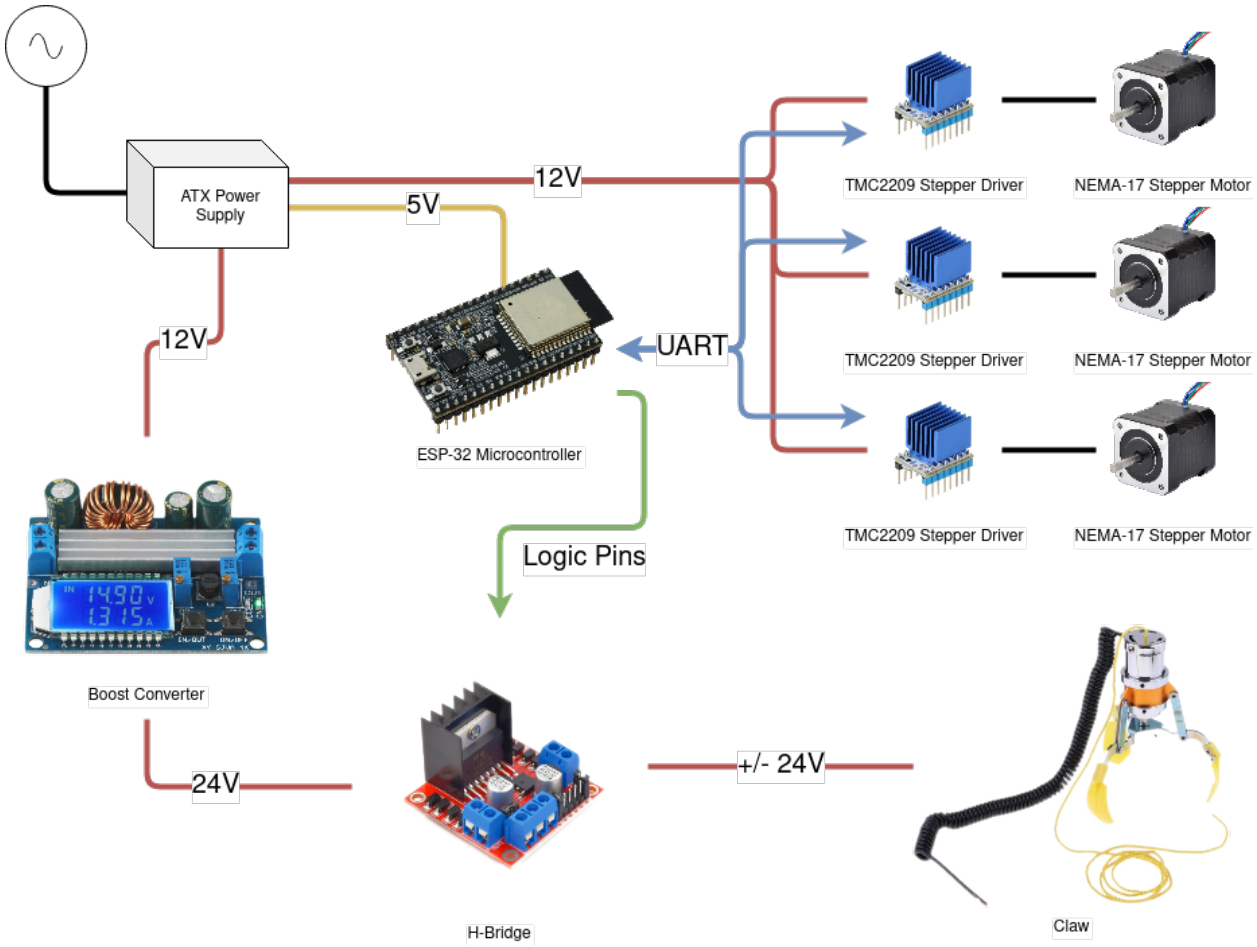


Figure 10: Low-level hardware wiring diagram

decided to use the Espressif ESP32. The ESP32 is a cheap and ubiquitous micro controller used throughout hobbyist electronics, and is also the chip of choice for many of WPI's own robotics classes. It is easy to interface with, and is compatible with the Arduino suite of software, making it a good choice when considering maintainability and extensibility.

For ease of use, the ESP-32 was soldered to a proto-board with screw terminals that allow for easy and solder-free installation and re-wiring. Extra GPIO screw terminals were included to account for future hardware expansion.

There is one future consideration that we had when selecting the ESP-32. While the micro-controller itself is widely used, the Instruction Set Architecture it implements is not open-source. This could prove challenging in the case of future, hardware-based challenges.

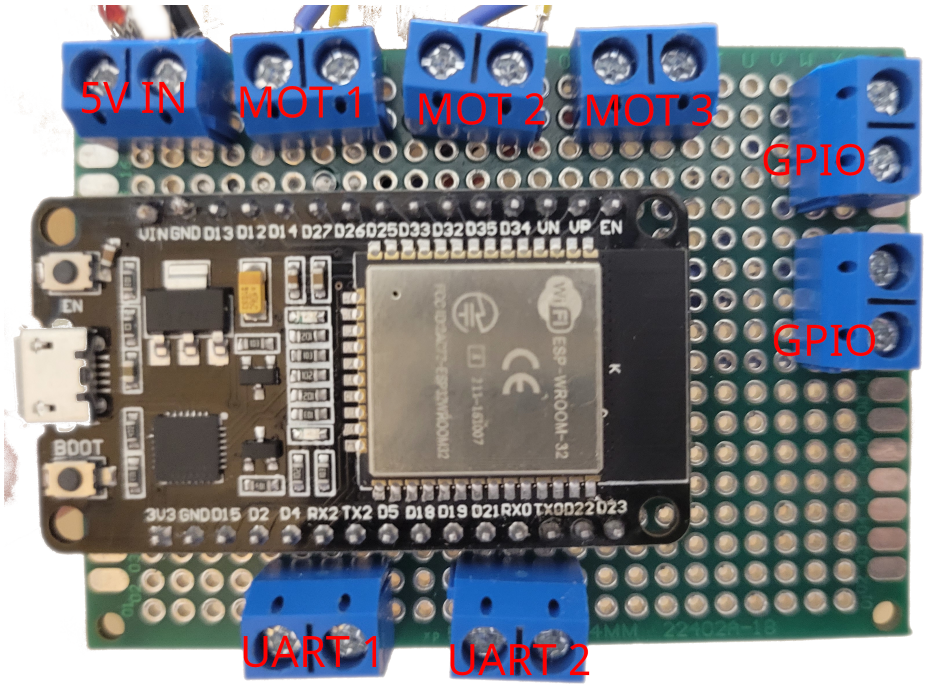


Figure 11: Soldered ESP-32 with input/output labels

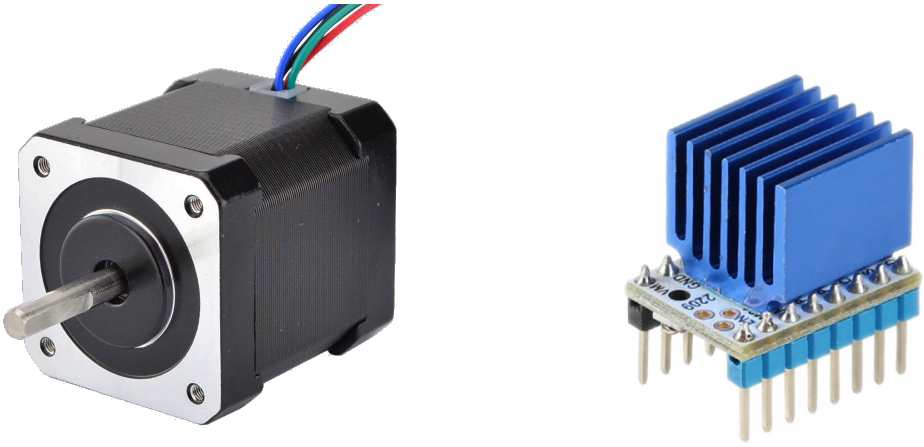
While these hardware challenges are out of the scope of this particular project, considerations were made with regards to the idea that future challenges may want to interface with the claw hardware directly. For this use case, a custom board was designed, which includes all motor controller and uses an open-source ARM based STM32 microcontroller in place of the ESP-32 board used for this design. More details about this design can be found in the Section 7, Future Work.

4.4.2.2 Stepper Motors and Drivers

To move the gantry system, we used NEMA-17 stepper motors. Stepper motors provide the advantage of being highly repeatable and providing good positional accuracy, which meant that encoders were not required to maintain the level of accuracy required to move the claw. The specific motors used in our case were 40mm NEMA-17 Bipolar stepper motors, able to provide up to 45Ncm of torque. This is a sufficient amount of torque to move our gantry system, as the friction in the system is very low - the bulk of the power goes into accelerating

the gantry, and the system does not require a very high amount of torque for the speeds that the gantry moves at.

One of the downsides, however, of stepper motors is that they require additional circuitry to be driven. We used the TMC2209 series of stepper motor drivers for this purpose.



(a) NEMA-17 Stepper Motor (b) TMC2209 Motor Driver

Figure 12: NEMA-17 Stepper Motors and TMC2209 Driver

These motor drivers are commonly used in hobbyist 3D printers, and can provide over 2A of current to the stepper motors, with the proper heat sink configuration. This is sufficient for the motors used by the claw machine, which can draw up to 2A, but generally draw well below 1A in their current configuration.

The TMC2209 series also provides a number of features beyond the ability to simply drive a stepper motor. One of these features is UART configuration and control. This allows the current limit, motor parameters, and advanced configuration to be done over a one-wire UART interface. They also provide motor stall detection, which is useful to detect when a motor has stopped moving because something is blocking it. This can be used, for example, to detect the bounds of the gantry and home the motors when the claw machine hardware first starts.

The drivers were soldered to proto-boards for installation into the machine, the

pinout of which can be seen below. They are also integrated into the possible future PCB, as found in the Future Work section.

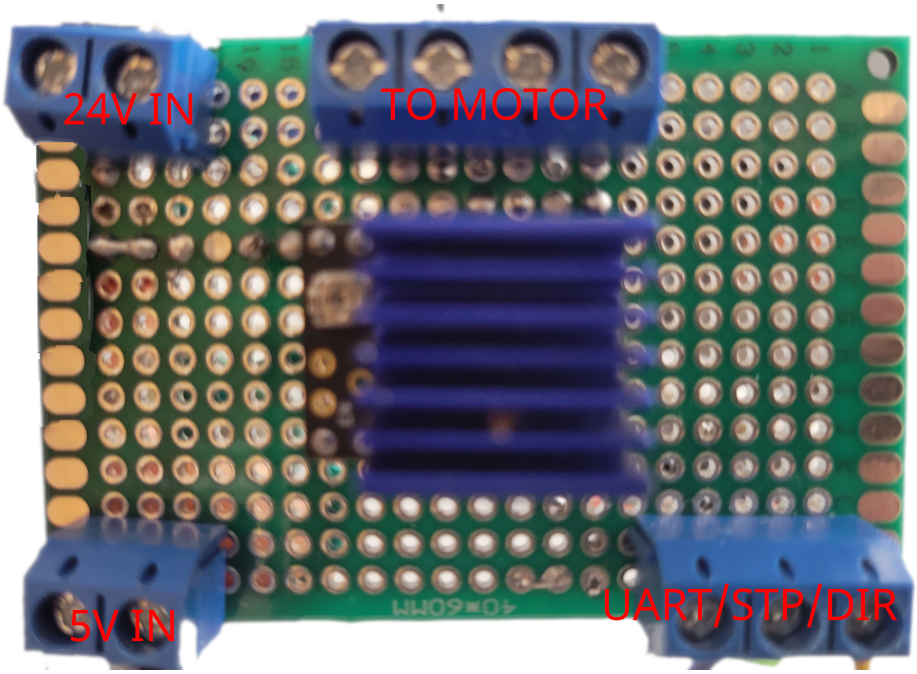


Figure 13: The annotated protoboard motor drivers

4.4.2.3 Claw Electronics

Figure 14: The claw used in the machine

The claw is an off the shelf electromagnetic claw, as would be used in a normal, commercial arcade machine. The claw by itself provides no advanced control, and is sim-

ply hooked up directly to a 24 volt power source. Since the claw mechanism is a simple electromagnet, it can be forced open or closed depending on the polarity of the incoming voltage. To control the polarity of the input voltage from the micro controller, a simple piece of circuitry called an H-Bridge was used, which essentially consists of four transistors and allows the high-voltage claw power supply to be controlled by the 5v logic pins on the ESP32.

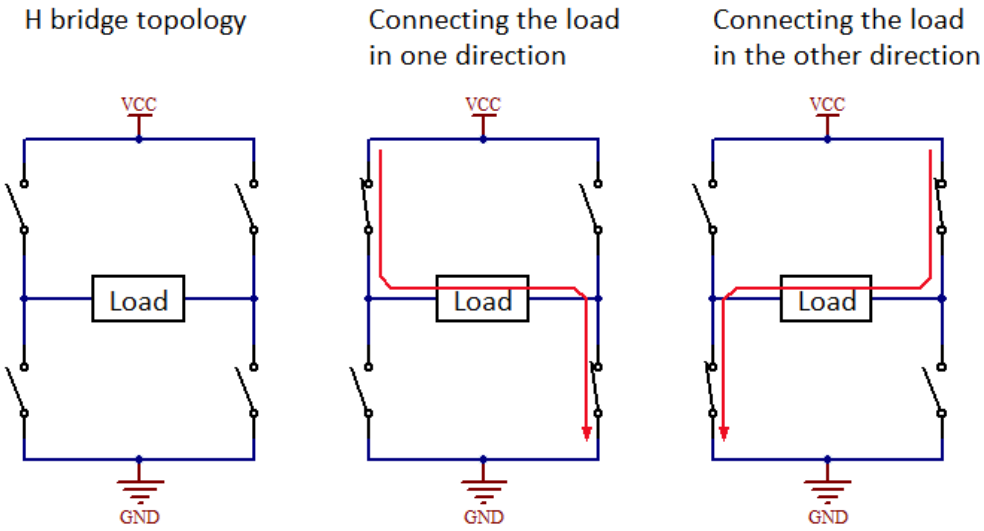


Figure 15: H-Bridge circuit diagram

As can be seen in figure 15, by alternatively powering different sets of transistors, the polarity through the load (in our case, the claw) can be reversed. The specific H-Bridge circuit used was the L298N, which is often used to drive DC Motors, and has the capability to drive the voltages and currents required to open and close the claw.

4.4.2.4 Power Supply

To power the low level electronics, a standard ATX power supply - like those used to power a home PC - was used. The power supply was selected as the majority of the low level electronics run on either 12 volts or 5 volts, two voltages which are standard on ATX power supplies. The only exception to this was the claw itself, which required a 24

Figure 16: ATX Power Supply

volt input. For this purpose, a boost converter was connected to boost the 12 volt output from the ATX power supply to the 24 volts required for the electromagnet in the claw. ATX power supplies also have the advantage of having built in cooling, and are generally rated for much higher wattage than the claw machine requires, meaning that this solution was a quick and easy way to power all the hardware required.

4.5 Embedded Software

In order to make the firmware on the ESP32 easy to maintain and expand, we decided to implement all of the low level control using the Arduino framework and libraries, using the PlatformIO extension to Visual Studio Code. Arduino libraries are available for the TMC2209 motor controller, which can be installed using PlatformIO.

The firmware on the ESP32 simply consists of a loop that waits for an activation signal over serial from the raspberry PI. Upon receiving the signal, the claw runs forward and backward, waiting for a button press from the user. Upon the first button press, the direction of the claw moves from forward-backward to left-right, and waits for another button press. Upon the final button press, the claw lowers, attempts to grab an item, drops any

item into the prize chute, and returns to the home position.

5 Design and Implementation of the Front End

Our Goals

The front end website of the claw machine had to do a lot of heavy lifting in this project. We needed to connect, manage, and serve custom pages to the claw machine. We also needed to serve an enjoyable CTF experience to our players. Some of our front-end development goals included:

1. Adapting PicoCTF functionality to suit our needs
2. Creating a consistent user experience across both the claw machine and the accompanying CTF website
3. Streamlining the account creation process and integrating with WPI's Azure AD SSO

5.1 Expanding upon PicoCTF

A significant number of changes were required to make PicoCTF suit our needs. We needed to integrate the physical claw machine into the user experience, so that they could seamlessly interact with the two. We also needed to add some features we wanted both for our own CTF, and some features that would be useful to the version of the CTF running in Software Security Engineering. Lastly, PicoCTF came with a lot of features that didn't make sense for our application, so we needed to properly remove those without affecting the rest of the infrastructure.

5.1.1 Claw Machine Integration

Our greatest distinction from the existing PicoCTF competition is the inclusion of the claw machine. This means that there are two ways competitors interact with our CTF: online, and at the claw. In order to unify these experiences, the website will also serve as part of the user interface for the physical machine. While users will access the site to submit challenges from their personal devices, scoreboards and other information will be displayed on a screen inside of the claw machine.

The existing scoreboards page of the website would not be suitable for the claw machine display. Like the rest of the site, it utilizes a horizontal layout, navigation bar, and relatively small text. To rectify this, we created a new scoreboard page that does not include the navigation bar, optimizes the layout for a vertical monitor, and uses large text and bright colors. This scoreboard lists the users in order of the number of points earned, displaying their username and point total in an arcade-game style layout. We used the same fonts and colors on this scoreboard as well as the main site, which further serves to unify the experience.

Another way we needed to integrate the machine and CTF platform was by rewarding students with attempts to use the claw machine via completing challenges. We did this by awarding them Claw Credits, which can be earned upon completing challenges that have Claw Credits enabled. Students can spend Claw Credits on a specific page on the site to activate the claw machine and try to win prizes. This helps further the connections between the claw machine and the CTF experience.

5.1.2 Addition of Administrative Features

One feature we implemented was the ability to assign a due date to a challenge. This would be useful for running Software Security Engineering's CTF platform, but would also help us run short-term CTFs that end after a few days. Due dates can be set under the Manage

Problems admin page, or in the `problem.json` file that controls metadata for each challenge. Once a due date has passed, players can no longer submit flags to the challenge. However, any points that students have earned by completing the challenge will still appear on the scoreboard, so they can still feel a sense of accomplishment from completing them. As a challenge approaches its due date, it will first change to yellow with a warning that it is due in 48 hours, and then at 24 hours it will change to red with a countdown until it is due.

Another set of functionality we desired was the ability to assign challenges to specific groups. For example, this allows us to have different challenges for normal users (in a "global" group), students taking Software Security Engineering, and members of the Cyber Security Club. These groups are created under the Classrooms page, as classrooms were an existing type of group in the PicoCTF infrastructure. We kept classrooms as a more restricted type of group that is not publicly joinable, and has a private leaderboard only visible to those in the group. Problems can be assigned to each group under the Manage Problems admin page, or in the `problem.json` file. Each group has different instances of each challenge with their own flag, different due dates, and different settings for if the problem is accepting flags or giving out claw credits.

Lastly, we included a feature to allow players to set a display name visible to others that would be different from their username. We would require that usernames be the same as a student's WPI username (if they had one). By allowing a separate display name, we can let students remain anonymous to their peers, use a preferred name if it doesn't match their username, or otherwise just choose a fun name. We hope this small feature would make students happy by allowing them to express themselves how they chose.

5.2 User Experience Design

Because this site is meant to accompany a physical machine, we needed to establish a clear vision for a shared aesthetic. This is important to ensure that the site and machine are

cohesive and easily identifiable. Early in our discussions of the project, we decided to adopt an "90's roller rink/arcade" design goal.

Figure 17: Moodboard created to refine the desired design aesthetic

We chose this style for several reasons. Primarily, we wanted a fun and eye catching design that would easily catch the attention of busy students on their way to class. However, we did not want the design to be overwhelming. The arcade color palette is bold yet familiar, making it a strong choice to satisfy those objectives.

This aesthetic is characterized by deep blue blacks, bright neons, and lots of glow. We wanted to capture this energy while still being accessible and easy to use. To achieve this, we decided that it would be best to preserve the overall structure and layout of the website, which is already easy to navigate. Instead of overhauling all of the React components, we simply added extra CSS to reach our aesthetic goals. The changes made were largely superficial, such as color and font choices. This means that the shape, spacing, and layout of the Bootstrap components remain the same. For color changes, we followed Material Design dark theme guidance [12] to ensure that there is sufficient contrast between background and text.

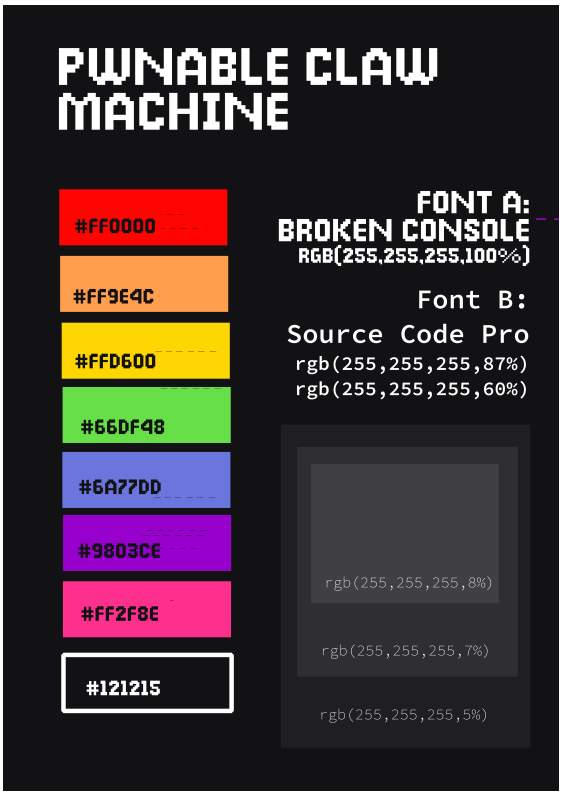


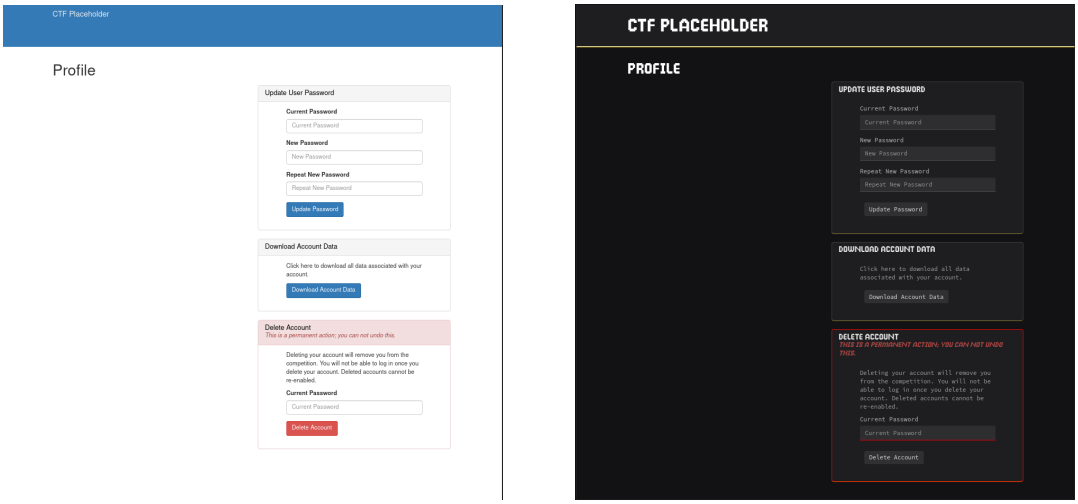
Figure 18: Guide to the colors and fonts used in the redesigned site

However, we did deviate from both Bootstrap and Material Design aesthetics when it came to fonts. We selected an 8-bit style font called Broken Console to be used for headings throughout the site. Its playful and retro appearance fits with our claw machine design goals, and helps unify the website with the machine. For non-heading text, we chose the monospace font Source Code Pro, which complements Broken Console nicely and is easier to read at smaller font sizes.

5.2.1 Story Tie-Ins

Additionally, the website needed to reflect the story-based decisions we had made (further discussed in Section 6). This meant that we needed to incorporate references to the plot and characters as well as callbacks to the arcade machine theming.

To incorporate those ideas, we created animated pixel-art sprites of our cat char-



(a) Original design (b) Updated design

Figure 19: Before and after site redesign

acters that appear sparingly throughout the site. Each cat character is associated with a specific type of challenge, and appears on information and warnings related to problems in that category. While originally we wanted to use easily recognizable internet cats, we found that copyright could be a potential issue [21][25]. For this reason, we decided to develop original cat images for use on our site.

We created four cat images, each with a unique personality and usage. Our first cat is a timid black cat that appears when a problem is about to expire. While black cats are typically thought to be bad luck, we hope his warnings will come in handy.

Other cats, such as our elegant seal point Himalayan, appear when users view the full description of a problem. These descriptions are phrased as if they were directives coming from the cats themselves, and serve to further reinforce the connection between the storyline and the CTF challenges.

These images were created using Piskel [6], a web-based pixel art tool. We chose to use Piskel because it is free and open source, and supports all the features we needed to create animated sprites of the cats. Piskel can be easily accessed from your web browser, and creating and exporting works is simple and free. Other popular tools, such as Aseprite [4],

Figure 20: Four cute cats

must be installed locally and cost money unless we were to build it from source. These two factors alone made Piskel a clear choice for the development of our sprites

5.3 Azure Integration

Aside from modifying the appearance of the site, we were also responsible for updating the login and account creation process for the CTF. PicoCTF is designed for competitors around the world, from different institutions. Their current registration process requires a large amount of information from participants, such as location, school, gender, and age. However, for our purposes, we can expect that practically all CTF participants will be WPI students. We wanted to integrate the login and account creation process with WPI existing Azure Single Sign-On (SSO).

By using WPI's SSO system, we lower the barrier to entry of the competition. It makes signing up to participate in the CTF as easy as checking your email, instead of filling out a somewhat-lengthy form. It also helps to associate user accounts with the students

that they belong to, so that we can more effectively gather demographic information in the future. Additionally, it ensures that students can only create one account to play. This is important because it reduces the risk of students creating multiple accounts to solve easy challenges repeatedly.

While we expect that that majority of students will register with SSO, we also wanted to allow individuals without WPI emails to compete. We expect that this functionality will mostly be utilized by high school students visiting WPI, or students from other universities. For this reason, when a user visits the site, they are prompted to choose between signing in with WPI SSO or going through the standard registration process.

In order to enable Azure SSO for the CTF, we had to register our app with Azure. Then, we added support for SSO into the CTF's front-end. Azure SSO uses the OpenID Connect (OIDC) protocol, which is an extension of OAuth 2.0 [11]. OIDC works similarly to OAuth in most ways, but it includes more data about the identity of the end user [9]. The key addition is the ID Token, which contains profile information about the end user such as their name and email address. This is in addition to the OAuth's Access Token, which grants users to access a specific resource [10].

To support this protocol, we decided to use the oidc-client-js library [14]. While it is currently no longer being developed, it is the most compatible with the existing PicoCTF codebase. Other libraries, such as Microsoft's MSAL.js [1] would be more difficult to build and save as part of our project. Since we cannot use any import statements, it was very important that a built, minified version of the package be readily available. While CDN-hosted versions of MSAL are available, that method of inclusion is inconsistent with other packages used throughout the site.

It is important to recognize that this implementation of SSO uses implicit flow to retrieve user data from Azure. Implicit flow is largely not recommended for many use cases due to security vulnerabilities. The data is returned via HTTPS without confirmation that

it has been received by the correct client, it is vulnerable to man in the middle attacks among others. However, it is not an issue for our use. Because we are using OIDC only to retrieve the ID Token, and not an Access Token, the risks are no longer significant [7][8]. However, it is a potential area of future improvement to convert the process to use Authorization Code Flow with PCKE instead of implicit flow.

In addition to adding front end support, we also created additional endpoints in the backend to register users who sign in with SSO. These methods are essentially small tweaks on the existing methods to support logging in, but with changes to the parameters. PicoCTF's original login API uses username to identify users, but for our case, we want to identify users by the email address retrieved from Azure. Since these users also do not have a traditional password, we updated the MongoDB schemas to list password as an optional parameter. However, we were careful not to remove any functionality so that the original PicoCTF endpoints will continue to work as expected, for users without WPI email addresses.

6 Story and Lore

One of the main goals of the project was to create an engaging and interesting CTF experience. It is our hope that making our CTF challenges engaging will lead to better learning outcomes and better user retention. One of the main ways we attempt to engage users in our CTF is via lore and a set of storylines. In the context of video games, "lore" tends to mean the backstory and details that accompany a main narrative [24]. In our case, we use it to mean the characters, setting, and other details that accompany the storylines we incorporate into CTF challenges.

We drew inspiration from CTF competitions run by the cybersecurity education company Hack The Box. Several competitions, such as their Universities CTF and Cyber Apocalypse CTF, contain lore and storylines that we have found fun and engaging. These

competitions center around a particular theme and backstory, upon which short storylines are built within categories of challenges. For instance, the 2022 Cyber Apocalypse CTF had an intergalactic theme, framing participants as a group of misfits coming together to overthrow a super villain who had escaped prison. By completing challenges, participants would work alongside various characters split between challenge categories, such as Miyuki for forensics challenges, or Bonnie for pwn challenges. Each of these characters had their own backstory with the villain, and thus their own storyline for players to engage with as they attempted to thwart the villain [3].

We also drew some inspiration from the yearly Advent of Code event. This event contains a series of 25 coding challenges released daily for the first 25 days of December. Each year, the event tasks players with helping Santa get ready to deliver presents by solving the coding challenges. While every year involves aiding Santa, the problems that Santa faces change year to year [28]. We've also enjoyed solving these challenges, and we've taken inspiration from the ways that Advent of Code continuously builds new storylines on the same basic premise. We carried this inspiration into designing our own lore for our CTF.

6.1 The Lore

The basis of our lore is a crew of four aliens that have crashed to Earth at WPI. These aliens are friendly (or ambivalent at worst), but often incompetent about human culture. They appear to humans as cats, because they have assumed that humans largely communicate via cat memes. There is also a shadowy Agency that is attempting to thwart or capture the aliens. The students are tasked with helping the aliens through various problems via solving CTF challenges.

A lot of the thinking behind our lore was inspired by cartoons such as *Road Runner* and *Tom and Jerry*. These cartoons have a main base of lore that small storylines are built atop of in the episodes. These storylines are often short-lived, in one or two episodes, so that

the audience can jump in at any point. Often in CTF competitions, some challenges stand on their own while others are completed in a series. Thus, we can have our own main base of lore, with small storylines that are told over one challenge or a small series of challenges, similar to cartoon episodes.

In order to make the lore of our challenges extendable by anyone creating challenges in the future, we decided to make some rules to govern the character interactions within the lore. These rules were inspired by the rules Chuck Jones created to guide the *Road Runner* cartoons [18]. Our set of rules for character interactions is as follows:

1. The Aliens will always at least slightly, and often humorously, misunderstand human behavior/the Students.

2. The Aliens will especially misunderstand human communication methods, and will choose methods of communication that are amusing, outdated, and not very effective.

3. The Aliens never mean the Students harm, and the Students never mean the Aliens harm.

4. Misunderstandings may mean that the Aliens and the Students briefly work against each other, but never in a way that is malicious or would harm anyone.

5. The Agency can mean to harm the Aliens, but often in a cartoonish sense. Think more "evil-laser-inator" than "kidnapping and experimentation."

6. The Agency does not mean the Students harm, but may attempt to thwart the Students' attempts to communicate/work with the Aliens.

The crew of aliens is made up of four characters, each with a different personality and role on the ship. They're also associated with different challenge categories, but can branch outside of those categories when the story requires. We wanted each character to be

easily distinguishable in both looks and personality to make the stories easier to understand with little introduction. The cats are described as such:

- This is Ensign Scar'dy's first time off-planet, and they're really trying to make it go well but they're very nervous about it. They joined the Exploratory Service because their friend said they should try new things, but they're a little skeptical. Ensign Scar'dy is really just trying to learn and get their footing without making anybody mad. They're a bit timid, but they're really good to have for General Skills challenges. They're also good to have for the Miscellaneous challenges, since they keep running odd jobs around the ship to make people like them. Ensign Scar'dy is a black cat.

- This is Clawde's second long-term mission, but their first after being promoted to engineer and they're super super excited about it! They joined the Exploratory Service because they love tinkering with spaceships, and someone has to keep the ships functioning on these long journeys. They LOVE learning new things, especially by taking them apart to see how they work. They sometimes struggle to understand people, and tend to prefer less-confusing machines. They're particularly handy for Forensics and Reversing challenges with their knack for taking things apart. Clawde is an orange cat.

- Princess has done a lot of stints on shorter missions, but this is her first long-term mission. It's starting to wear on her a bit, but she is doing her best to stay above it all. She joined the Exploratory Service as the latest in her family line of communications experts. She's very good at her job, but she's also very used to getting what she wants. Her encryption skills make her the go-to for Cryptography challenges, and her adept ways with a social media presence make her great for Web challenges as well. Princess is a fluffy Himalayan cat.

- Captain is a lifelong explorer who has been with the Exploratory Service for 31 years. He joined because it was kind of the thing everyone did where he grew up, but he

came to love it. This is his fourth long-term mission, second as the Captain. He enjoys looking after a crew and helping them explore the cosmos. He's seen many things in his time, so he can be gruff or stern, but he wants to see the universe and get everyone home safe. In his spare time he likes tinkering with the ship's circuits and the kernel it runs, so he's the guy to go to for Pwn or Hardware challenges. Captain is a white cat with a space helmet and some facial scars.

Lastly, the Agency is a shadowy group of ne'er-do-wells meant to be the antagonists of the story. It is our thinking that having an antagonist in some storylines will help the players feel like they are affecting the story when they help defeat the Agency. The Agency's goal is to investigate and thwart the aliens. They do not need to show up in every storyline, since there are other ways to create stories with interesting conflicts, but they are a good narrative tool at our disposal.

6.2 Story Incorporation

Hack The Box also helped inspire the ways that we integrated our story into the website and challenges. Hack The Box typically relayed storyline information via challenge descriptions, which are short pieces of text that explain the technical premise and lore connections of a challenge. They also included some small pieces of lore within files given for a challenge, such as visual elements or pieces of text [3][5]. One thing we appreciated about Hack The Box's implementation of their story is that it did not distract from the technical aspects of the challenge. You could pay attention to the story if it interested you, but it was also possible to minimally interact with the lore and still solve challenges.

Based on the techniques we observed from Hack The Box competitions, we made ourselves some guidelines about our implementation of the story:

1. The user should not be required to engage with the story in order to solve challenges.

2. The story will be told mostly through challenge descriptions, with some minor flavor text within the challenges themselves.

3. The story portion of challenge descriptions will stay short at about 2-5 sentences.

4. The challenge description will connect the technical aspects of the challenge to the lore.

6.3 Applying the Story

We decided to create storylines to apply to existing Software Security Engineering CTF challenges, rather than make our own challenges from scratch. The challenges we created lore for were separated by the type of vulnerability they teach. This included stack vulnerabilities, format string vulnerabilities, and heap vulnerabilities. We also created a story for The Bar, a challenge incorporating many vulnerabilities learned in the course that students must complete to pass the class. Lastly, we created an introduction to the lore for the class, and story for two oral exams that students have to complete to receive B and A grades.

Our goal with this set of storylines was to introduce the characters, the cats' spaceship, the claw machine, and why all of them were on Earth. We first needed to introduce how the spaceship had crashed to Earth, so that CTF participants would understand why they were interacting with the cats. The ship had crash-landed on Earth after the aliens had mistakenly set their navigation to disregard dwarf planets, and thus they hit Pluto on their way through our galaxy. We alluded to this by creating a short message from the Agency that would go in the News section of the CTF site, as shown in Figure 21.

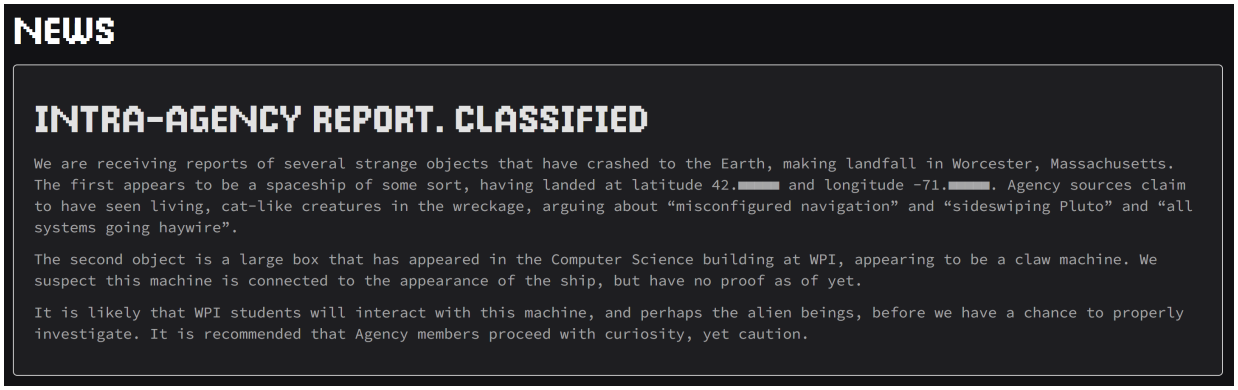


Figure 21: Agency Report to introduce the story to students

Each of the storylines associated with a series of challenges focused mostly on one character. The next storyline was associated with the stack vulnerability challenges. This gave a brief introduction to each of the aliens one by one via helping one of them, Clawde, fix various parts of their ship. This gave some insight into the attitudes of each of the cats (a timid Ensign Scar’dy, a bossy Princess, and a stern but caring Captain), and also gave us more time with Clawde. An example of one of these descriptions is shown in Figure 22.

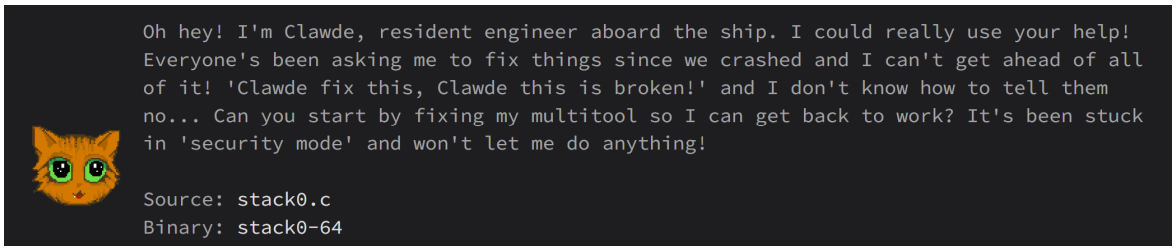


Figure 22: Challenge description for stack0-64

The next section of the story was for the format string challenges, and gave further insight into Princess. The players would help her fix her communication relays by completing challenges. It also played up the idea that the aliens look like cats, but are not actually cats, as Princess is perplexed by cat noises coming from her translation matrix:

Description for format1r-64

I can hear voices now, but they all sound strange. I’m hearing mews and yips instead of words...

I can't understand any of this. The matrix must be set to the wrong output language. Can you fix it? You'll need to use a format string to overwrite a global variable in format1-64.

Students would then go on to solve a series of heap vulnerability challenges, which gave further insight into Ensign Scar'dy as well as introducing the claw machine into the story. We envisioned the claw machine as a drone belonging to the aliens, used for picking up fuel and other supplies. This storyline involves fixing up the claw machine so it collects the correct fuel. Here is the challenge description that introduces Ensign Scar'dy and the claw machine:

Description for heap0r-64

Wh-who's there? Oh wait, you're the one who's been helping out! I'm Ensign Scar'dy, I just do a bit of everything around here. Right now, Captain has asked me to make sure the Collection Logistics Automated Workhorse drone is working... We usually just call it the CLAW machine. It's really important that we get it running so it can collect fuel and other supplies for us, but it will only run an environmental scan right now. Can you help me exploit some heap-based bugs to run more routines?

The last challenge we applied lore to was The Bar, the challenge that students are required to complete to pass the class. This challenge introduced Captain, who asks the player to fix the life support system on the ship so it can leave if so desired. This is the last thing that needed to be fixed on the ship, and thus is the last challenge required. The description is as follows:

Description for This is The Bar

Hello, engineer. I am Captain on this vessel. You've done a lot of work to get this ship back in order, and my crew speaks highly of you. I would like to ask one more favor of you. Our life support

system has gone into lockdown, specifically the Barometric and Atmospheric Regulation, or BAR. It has activated several protections that are preventing us from using it, and thus preventing us from re-entering space. Please combine the skills you've used in previous challenges to bypass the default x86-64 protections in the BAR and get our life support back online.

New to the course this year is a set of oral exams, in which the student walks the professor through a solution to a problem of the professor's choice. There is one required to get a B in the course, and one requiring an explanation of a more difficult problem required to get an A. We created descriptions for these oral exams, in which Captain is asking the student to explain these problems so he can ensure the quality of their work and thus the safety of his crew. The B-level exam (seen below) has Captain concerned with baseline safety of the student's repairs, and the A-level exam has Captain ask about the longevity of the repairs.

Description for B-level Oral Exam

You've been a great help to my crew in our ship repairs, but I want to make sure your work is up to snuff. Is your work safe for my crew? Or have you been making shoddy patches that will break when we attempt to take off? I have a way to ensure the quality of your work. I would like you to discuss one of your intermediate repairs with your CS Department's Intergalactic Ambassador, Professor Walls. He will choose the challenge to discuss, and you will need to walk him through the script you used to repair my ship. I hope he deems your work adequate for the safety of my crew.

As this was our first attempt at a set of storylines, they didn't incorporate all of the rules we set out for character interactions. The rules govern some of the ways that we can create conflict in a particular storyline. However, for an introduction to the characters and story, we decided to forgo some aspects of the lore, such as the aliens being inept at communication with humans, for a more understandable storyline. These aspects can be

re-incorporated in later storylines once students are more familiar with the characters.

7 Future Work

As this was the first team to work on the Pwnable Claw Machine, we have some aspects of the project that we had to leave for future teams. We summarize those in this section.

7.1 Pwnability of the Claw Machine

Our original idea for the claw machine pictured students actually being able to hack the claw machine itself. We imagined that having a claw machine that you were allowed to tamper with and hack would be intriguing to students, as well as a fun introduction to hardware security. It would also help our CTF be unique from other competitions that students could compete in online. While we focused more on the infrastructure to support the CTF and claw machine, we hope that future teams would make the claw machine actually pwnable.

7.2 Audience and Content

We originally planned to create our own challenges for this CTF, before eventually pivoting to applying storylines to existing challenges. We spent some time analyzing the audience that would participate in the CTF to decide what content would be interesting and helpful for them to learn via our challenges. We also spent time analyzing other CTF platforms to get an idea of what types of challenges they host.

7.2.1 Audience

Our users will mostly be students in and around Fuller Commons, or any other locations where we may place the claw machine. These students can be divided into two categories:

Inexperienced Students: These students would mostly be first and second year students. They are largely unfamiliar with cybersecurity in general, and may or may not have knowledge of cybersecurity offerings at WPI. They may need encouragement to become interested in the CTF, which is where things like prizes from the claw machine and our storylines will come in. These students can be intimidated by CTF challenges that seem too difficult, so they may need more guidance to get started.

Experienced Students: These students are more likely to be third and fourth year students. They are familiar with CTFs, and are likely aware of cybersecurity at WPI. They are more likely to be interested in difficult challenges, and may need less encouragement to get started with the CTF. Prizes and lore would be more of a bonus than a necessary incentive.

Our main audience will be inexperienced students, because freshman and sophomores have limited points of entry to cybersecurity at WPI. Providing them with a fun and rewarding entrance to the field is our primary goal. While our main audience is inexperienced students, it is in our interest to engage more experienced students as well to drive collaboration between experienced and inexperienced students and increase word-of-mouth advertising.

We also have to consider what relevant classes inexperienced students have taken in order to establish a starting point for learning. The WPI CS department recommends that freshmen take:[15][16]

- Intro to Program Design (CS1101 or 1102)
- Object Oriented Design Concepts (CS 2102 or 2103)

- Then EITHER
  - Systems Programming Concepts (CS 2303)
  - Machine Organization and Assembly Language (CS 2011)
- OR
  - Discrete Mathematics (CS 2022 / MA 2201)
  - Algorithms (CS 2223)

Thus, for our inexperienced students, we can assume basic programming knowledge in Java. From there, most students will either have a deeper understanding of Java programming (through Algorithms), or some C-programming knowledge and assembly experience (through Systems and Machine Org). Some freshmen may have some combination of the two paths. Most students will have caught up with the other path of classes by the end of their sophomore year, if not the fall semester.

7.2.2 CTF Content

Given our analysis of the audience for our CTF, we can then start deciding what level of difficulty we would have challenges be, as well as what types of challenges we would include..

As our intended audience is freshmen and sophomores with little to no cybersecurity skill, we will need to have mostly easy to medium difficulty challenges. There should even be an abundance of easy challenges, since the point of this CTF is to teach people when they aren't being reached by our cybersecurity clubs or classes.

However, there is room for challenges of all difficulties. We can have some harder challenges for people to work on after they've mastered the basics. There's even some incentive to have some expert difficulty challenges since there is no time limit on the availability. Many CTFs only run for a few days to a week, but since ours will likely be indefinite, it

doesn't matter so much if difficult challenges take many hours to work through. We can have a select few challenges that are extra difficult for those well-versed in cybersecurity to give them a sense of accomplishment.

To create a list of challenge categories we would like to have in our CTF, we looked to Hack The Box, PicoCTF, and National Cyber League's practice gym for inspiration. Both PicoCTF and National Cyber League's practice gym are long-term CTFs meant to help people learn. They host a wide variety of challenge categories and difficulty levels [23][19]. Hack The Box is much more difficult, but also has the infrastructure to host a wider variety of challenges that are worth considering [2].

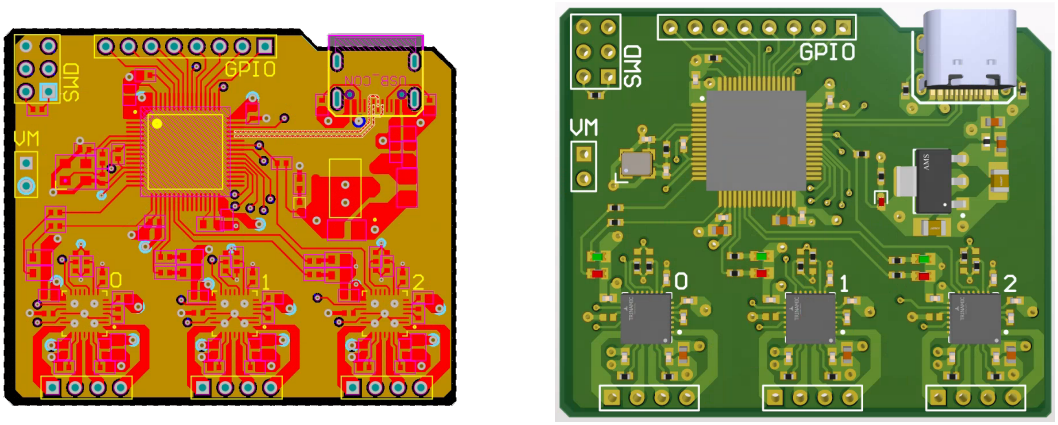
Most CTFs include challenges in Pwn, Cryptography, Forensics, Reversing, and Web categories, so ours should as well. This provides a wide array of topics for people to engage in and learn. Less commonly seen is a Hardware category, but we would like to include it since having a physical claw machine allows for unique opportunities to create physical challenges. We would also like to take after PicoCTF and include a General Skills category, which would teach basic command line and data format skills. These topics are usually considered prerequisites for many CTF challenges, but there often isn't an avenue to learn them. This section could also teach various command line skills that are not explicitly taught in WPI's curriculum. Lastly, most CTFs include a Miscellaneous category for the challenges that don't really fit elsewhere, and we'd like to include this as well.

7.3 Integrated Control Board

One of the considerations we had when selecting suitable control hardware for the claw machine was whether it would be able to be easily integrated into future hardware based challenges. A large component of hardware based challenges is often binary exploitation, or "pwn"-style challenges. These types of challenges often require a deeper understanding of underlying hardware, especially the ISA (Instruction Set Architecture) of the platform that

is being exploited.

This poses a problem, as the current ESP-32 micro controller that is used to control the hardware has a closed source ISA, making exploitation challenges significantly harder to both design and play. For this reason, and because a single PCB board is a significantly cleaner solution to the internal wiring of the claw machine, a potential integrated control board was designed that implements an STM-32 ARM based microcontroller. ARM is an open source ISA, and is directly applicable to many real-life cyber-security situations today.



(a) PCB - Layer View (b) 3D View of PCB

Figure 23: Prototype PCB Design

The PCB design was done in Altium Studio. The design is a fairly standard 4 layer design, but special consideration was given to motor power routing, given that the board could see up to 6 amps of current when supplying the motor controllers. The micro controller is a Cortex M4 STM32L475 from ST Microelectronics, and the three motor controllers are all the Trinamic TMC2209 integrated stepper motor drivers. The board also contains logic power circuitry, and a USB Type-C connector for power and integrated serial communication. For future iterations of the board, it would also be desirable to include boost converter and H-Bridge circuitry for the claw, as to make the board a truly all-inclusive control unit for the machine.

8 Conclusion

Over the course of the 2022-2023 academic year, our team successfully built a physical claw machine, made significant modifications to the PicoCTF infrastructure, and developed a overarching story for the CTF. The physical claw machine provides a fun and engaging way to reward students for participating in our CTF, and provides attractive advertising to increase awareness of our CTF and cybersecurity at WPI. The PicoCTF infrastructure has been improved with a full visual redesign in the style of 90's arcades and internet cats, and numerous quality of life features have been added to improve the experience for players and administrators alike. Lastly, the overarching story behind our CTF creates a fun narrative for players to interact with throughout the CTF, hopefully increasing education outcomes and user retention. We hope that the Pwnable Claw Machine will be a delightful way for students to engage with cybersecurity at WPI.

References

[1] AzureAD. Azuread/microsoft-authentication-library-for-js: Microsoft authentication library (msal) for js.

[2] Hack The Box. All about hack the box.

[3] Hack The Box. Cyber apocalypse ctf 2022 - intergalactic chase.

[4] David Capello. Asesprite.

[5] ctftime. Cyber apocalypse ctf 2022: Intergalactic chase.

[6] Julian Descottes. Piskel: Free online sprite editor.

[7] Auth0 Docs. Implicit flow with form post.

[8] Auth0 Docs. Implicit flow with oidc.

[9] OAuth 2.0 Documentation. Id tokens vs access tokens.

[10] OAuth 2.0 Documentation. What are oauth access tokens.

[11] OpenID Foundation. Openid connect, Nov 2022.

[12] Google. Material design dark theme.

[13] Hannari_eli. *Tokyo Japan July 28 2019 Ufo Stock Photo 1465015262*. Shutterstock, Jul 2019.

[14] IdentityModel. Identitymodel/oidc-client-js: Openid connect (oidc) and oauth2 protocol support for browser-based javascript applications.

[15] Worcester Polytechnic Institute. Computer science – the first-year experience, 2023.

[16] Worcester Polytechnic Institute. Undergraduate academic catalog, 2023.

[17] Rohit Jha. Introduction to 'capture the flags' in cybersecurity, Jun 2020.

[18] Chuck Jones, Matt Groening, and Steven Spielberg. *Chuck Amuck: the life and times of an animated cartoonist*. Farrar Straus Giroux, New York, NY, 1989.

[19] National Cyber League. About.

[20] Mozilla Developer Network. Getting started with react.

[21] Scott Neuman. Grumpy cat awarded $710,000 in copyright infringement suit, Jan 2018.

[22] Stack Overflow. Stack overflow developer survey 2022.

[23] PicoCTF. About picoctf.

[24] Pablo Seara. Telling stories: The importance of lore in video games, Nov 2016.

[25] Katie Van Syckle. Keyboard cat and nyan cat come out ahead in lawsuit against warner bros.., Sep 2013.

[26] Carnegie Mellon University. Picoctf.

[27] Carnegie Mellon University. Picoctf/picoctf: The platform used to run picoctf 2019.

[28] Eric Wastl. Advent of code 2022.