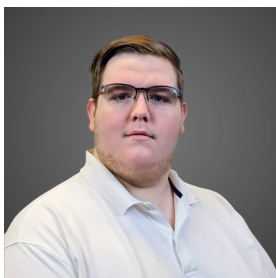


# Cyber Arena: Tools of the Trade

Benjamin Miller  
Advisor: Dr. Phillip Huff

March 2024





The Cyber Arena is a teaching tool that assists students with learning cybersecurity concepts in a virtual environment. For example, one of the labs the Cyber arena developed is ransomware, which runs a box with a simulated ransomware attack. The student must go through the attack without any defense and then set up defenses, such as backup files, against this attack so that it doesn't go off and mitigates the damage, if it did any. In doing so, the student learns about the attack and how to think like an adversary and defend against the threat. This tool is developed by a team of

people, and just as construction workers have tools in their toolbox, the Cyber Arena team uses specific software and resources to help develop and improve the teaching tool we have created. This 4 part series will run through the tools the development team uses and why, briefly explaining how the Cyber Arena is developed. One of the tools the team uses is an integrated development environment (IDE) called PyCharm[1]. It allows the team to code the front and back end of the Cyber Arena utilizing a variety of languages and extra tools. The front end of the code is the visual aspects, while the back-end is the base code and functionality.

An IDE allows a developer to create, modify, debug, and run the code developed and helps immensely with creating pages and specific back-end functions in the Cyber Arena. PyCharm helps put all of the coding resources in the same spot. PyCharm also allows everyone to work out of the same environment so that if the development team has an issue, they can quickly assist without worrying about the tool having different looks or another problem like this.

The Cyber Arena team does code and markup in several languages, including HTML, CSS, JavaScript, and Python[3]. The HTML, CSS, and JavaScript languages are primarily used for the Cyber Arena website functionality. The team uses programming languages to make the website facing side function correctly and helps to make the website look good. The back-end functionality of the Cyber Arena is primarily coded in Python, with the admin scripts, as well as much of the basic functionality, such as creating an arena, starting, stopping, and deleting, being handled by Python scripts through Google Cloud Services.

Jinja[2] is also a tool that is used in PyCharm. Jinja is a web template engine for Python, allowing the team to create routes for

the site and use some variables that come from the datastore, a piece of Google Cloud Services. It allows for more complex variables. For example, in the student webpage, a Jinja variable called dns suffix is used to grab the DNS suffix of the page for that specific arena and make the guacamole server web page, which is the front-facing part of most of the workouts, available to click into. It is also required for running Flask[4] applications.

Flask is an application that is a web framework that allows the set up of the arena webpage locally. This means the developers can run a Flask configuration in PyCharm and access the Cyber Arena without changing the public-facing site. This helps debug and allows a look at changes as the development team makes them.

Lastly, two settings come specifically from PyCharm. Configurations and debug mode. Configurations in PyCharm allow developers to run specific scripts or Flask applications from PyCharm. Debug mode in PyCharm allows the user to set stop points and look at variables and other issues in the code. It will enable a user to step through every step of the code to see where the problem is so it can be fixed much more effortlessly.

The Cyber Arena uses PyCharm in developing, testing, debugging, and improving the tool. The development team uses PyCharm to code in HTML, CSS, JavaScript, and Python. They use Jinja and Flask to help tie the code to a web application. They use PyCharm's inherit configurations and debug mode to test and fix bugs. The PyCharm IDE is not just a wrench in the toolbox but a multi-tool that helps in various ways. It also helps tie the application together with Guacamole, Google Cloud Services, and GitHub, the team's other big tools in development and testing.

## References

- [1] Getting started: PyCharm. PyCharm Help. (n.d.). <https://www.jetbrains.com/help/pycharm/2023.3/getting-started.html>
- [2] Jinja. Jinja. (n.d.). <https://jinja.palletsprojects.com/en/3.0.x/>
- [3] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.
- [4] Welcome to flask. Welcome to Flask - Flask Documentation (3.0.x). (n.d.). <https://flask.palletsprojects.com/en/3.0.x/>