OntarioTech Science

vialab
visualization for information analysis

Athena CTF: A Modular Framework for Instructional Capture-the-Flag Challenges

Zach Frank

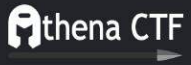
Supervisors: Randy Fortier, Mariana Shimabukuro

What is a Capture the Flag (CTF)?

- A collection of cybersecurity challenges
- Each challenge involves one or more vulnerabilities
- Team or individual based competition
- Leaderboard and prizes

Problem with current CTFs

- Building CTF's requires significant time and expertise
- Difficult to deploy and manage
- Static flags lead to cheating
- Static or no hints can lead to a bad user experience
- The learning ends when the competition ends



Home | How To Play | Documentation | Start | Dashboard | Leaderboard

Welcome

How To

Start Now | How it works

Included Vulnerabilities

- Broken Access Control
- Broken Object Level Authorization
- Cryptographic Failures
- Authentication Bypass
  Coming Soon!

Recommended Tools

- ZAP
  Intercepting requests, fuzzing and scanning
- BurpSuite
  Similar to ZAP, free version is missing features
- Python/JS
  Automating tests, capturing data
- CrackStation
  Online hash database for fast cracking
- HackTricks
  Common exploits for a variety of attack surfaces

Stakeholders Needs

Professors

- Fast development
- Low resources

CTF Administrators

- Need many levels
- Limited hosting resources

Other Educators

- Support for hints
- Support for static flags for learning

Context Driven Hints

Hints based on

- Past user submissions
- Static solution provided by admin

Multi-LLM* support

- Claude
- ChatGPT
- Ollama (Any model)

*Static hints are also available for those who choose not to use an LLM

Simple Level Creation

- HTML (Jinja2) templates provided
- Python functions for objects
  - Buttons · Forms · Tables · Menus
- Supports safe and unsafe HTML code
- No click database setup with Beanie ODM

Other Features

- Shareable Containerized Competitions
  - Each competition can be containerized with Docker
  - Users can download and complete past competitions
- Offline Admin Mode
  - Upload and Download data with CSV and JSON

Expert Evaluation

- 45-minute pair-programming session
- Participants had not read any documentation
- One TA and one professor

Expert Evaluation Feedback

Positive Comments

- The framework was simple to use
- The framework yielded clean and useable results
- The framework provided adequate tools for production use

Suggestions

- Docstrings on framework internals **COMPLETE**
- Restructure framework for easier level addition **COMPLETE**
- Add more helper functions for things like logins **PARTIALLY COMPLETE**

In Lab Study

Educator

- TA built 3 levels of varying difficulty
- Documentation was provided, but no pair programming

Students

- Students completed pre and post surveys
- Completed in the final week of labs

REB #32369

In Lab Study Results

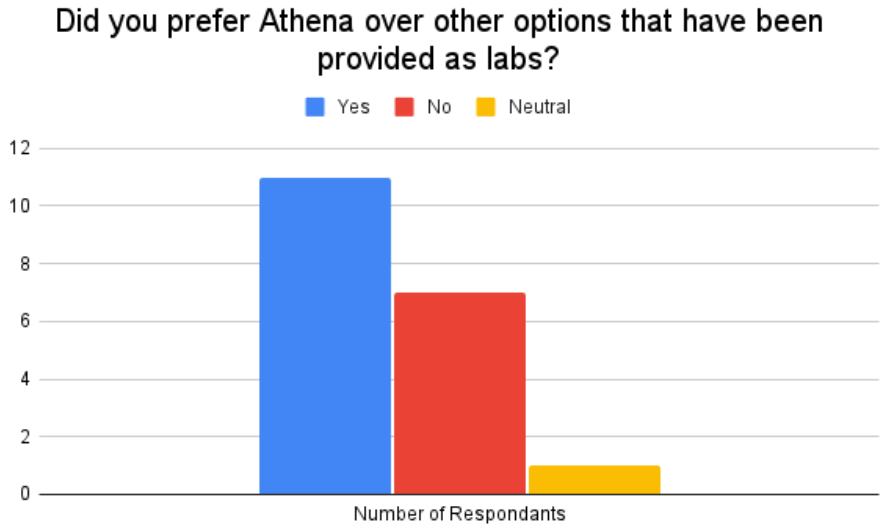
Educator

Positive Comments

- Level building was simple and fast
- Documentation, including docstrings, was good

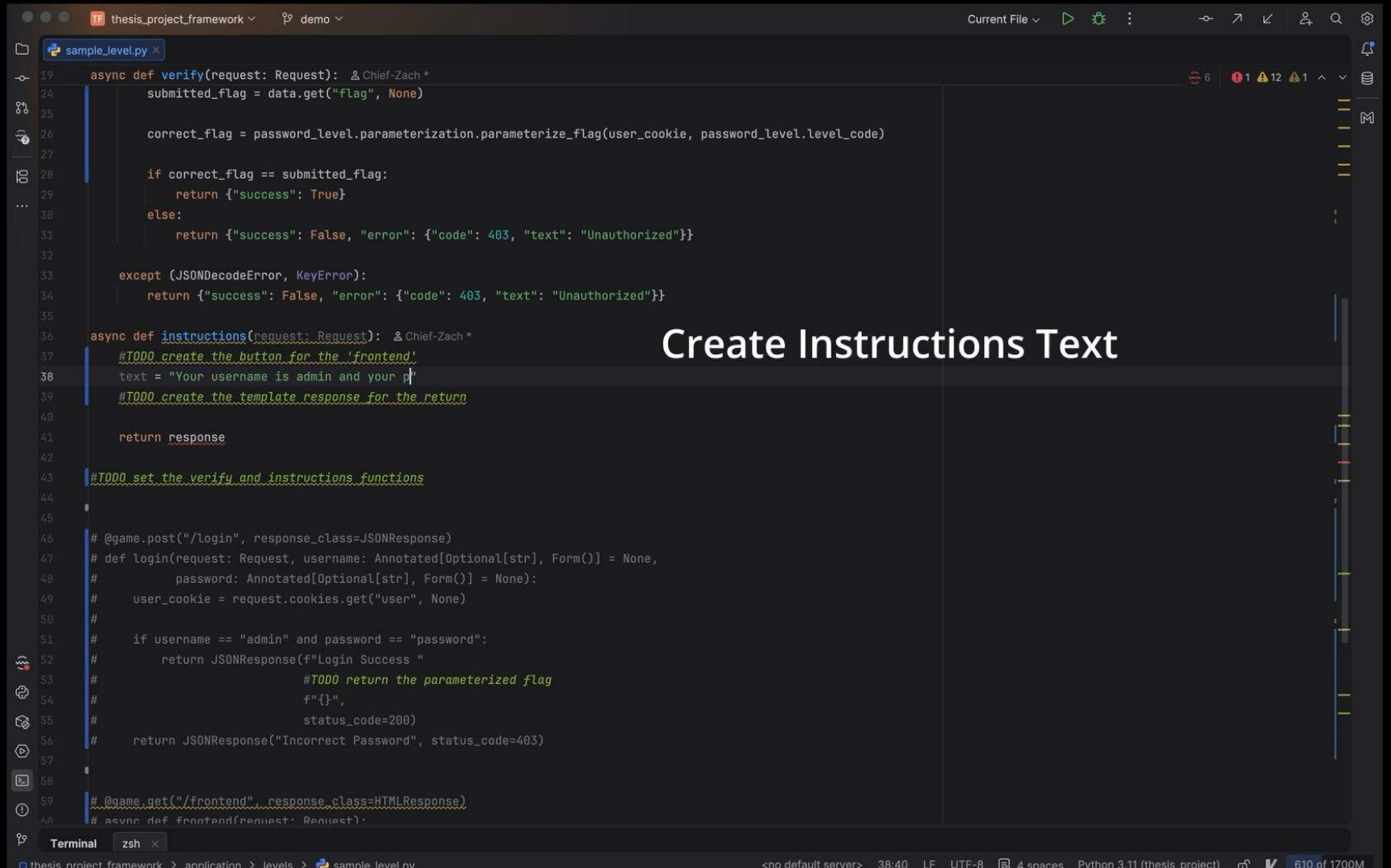
Student Suggestions

- Hints could be more specific
- Hint load times were too long



Future Work

- Adding more LLM options to the hint class
  - Gemini
- Adding more helper functions
- Creating more levels
- Modifying the prompt for better hint results
- Addition of Kubernetes for safer deployments


thesis_project_framework
demo
Current File
sample_level.py
async def verify(request: Request):
Chief-Zach *
submitted_flag = data.get("flag", None)
correct_flag = password_level.parameterization.parameterize_flag(user_cookie, password_level.level_code)
if correct_flag == submitted_flag:
return {"success": True}
else:
return {"success": False, "error": {"code": 403, "text": "Unauthorized"}}
except (JSONDecodeError, KeyError):
return {"success": False, "error": {"code": 403, "text": "Unauthorized"}}
async def instructions(request: Request):
Chief-Zach *
#TODO create the button for the 'frontend'
text = "Your username is admin and your p"
#TODO create the template response for the return
return response
#TODO set the verify and instructions functions
@game.post("/login", response_class=JSONResponse)
def login(request: Request, username: Annotated[Optional[str], Form()] = None,
password: Annotated[Optional[str], Form()] = None):
user_cookie = request.cookies.get("user", None)
#
if username == "admin" and password == "password":
return JSONResponse(f"Login Success "
#TODO return the parameterized flag
f"{}",
status_code=200)
return JSONResponse("Incorrect Password", status_code=403)
@game.get("/frontend", response_class=HTMLResponse)
async def frontend(request: Request):
Create Instructions Text
Terminal
zsh
thesis_project_framework > application > levels > sample_level.py
<no default server>
38:40
LF
UTF-8
4 spaces
Python 3.11 (thesis_project)
610 of 1700M

OntarioTech Science

vialab
visualization for information analysis

https://thesis.zachfrank.dev

Athena CTF: A Modular Framework for Instructional Capture-the-Flag Challenges

Zach Frank

Supervisors: Randy Fortier, Mariana Shimabukuro