Flag Gearup
Learn about web security by attacking a broken, unknown website:

• Poke around the site to figure out how it works
  – You don't access to the code! Learn about the system by testing

• ... then break it!

• After that, write vulnerability reports about each vulnerability

• CS1620/CS2660: Additional, multi-step attack: Bob's Router
# The assignment

- Find and write up at least four (4) vulnerabilities
- Each must be from a distinct *vulnerability category*
  - Can't count the same category more than once

<table>
<thead>
<tr>
<th>Category 1</th>
<th>Category 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad Password Hashing</td>
<td>Insecure Direct Object Reference</td>
</tr>
<tr>
<td>Business Logic</td>
<td>Path Sanitation Bypass</td>
</tr>
<tr>
<td>Client-Hidden Sensitive Data</td>
<td>Referrer-Based Access Control</td>
</tr>
<tr>
<td>Cookie Poisoning</td>
<td>Reflected XSS</td>
</tr>
<tr>
<td>Cross-Site Data Access</td>
<td>SQL Injection</td>
</tr>
<tr>
<td>Cross-Site Request Forgery (CSRF)</td>
<td>Session ID Prediction</td>
</tr>
<tr>
<td>File Inclusion</td>
<td>Session Fixation</td>
</tr>
<tr>
<td>File Upload</td>
<td>Stored XSS</td>
</tr>
<tr>
<td>HTTP Parameter Pollution</td>
<td>UI Redress / Clickjacking</td>
</tr>
</tbody>
</table>

Haven't heard of some of these before? Don't worry, we have resources to help!
The Wiki

We've provided each wiki that explains each vulnerability in detail
• Find it here: https://cs.brown.edu/courses/csci1660/wiki/

Use the wiki to...
• Learn about each type of attack and how it works
• See "Criteria for Demonstration" => what you need to show us to count as a vulnerability
• Find more references for further reading
How you'll work on the project

- "Flag portal container": download a container on your system
  - Similar to dev environment from Project 1
  - Hosts website for you to attack
- Use (almost) any other tools on your computer
  - "Developer tools" in your browser (Firefox highly recommended)
  - Your dev container from Project 1 (for Linux tools, running scripts, etc.)
  - Burp suite
  - Anything else as long as it doesn't automatically find vulnerabilities for you

You won't be writing a lot of code—most of your time will be trying out things, maybe making small code snippets/scripts, etc.
How to get started

Project setup guide: https://hackmd.io/@cs1660/flag-setup-guide

What's in this guide
• How to update your dev container/Docker setup from Project 1
• How to clone the Flag container
• Helpful resources if things go wrong with the containers
About the container environments

• Flag uses a new container, separate from your "dev container" from project 1
  – Bob's router has one more container
    ⇒ 3 total

• Interact with new containers with a script called run-container that will do most things for you
  – Run it like you would use cs1660-run-docker
    - Download Container Image
    - Run it
    - Reset it back to original state
Important container terminology

- **Container image** ("image"): read-only package of the files/settings for how the container runs
  
  => You download from us.

- **Container instance** ("container"): created when container started, read-write
  
  => CREATED WHEN YOU RUN ./run-container
  
  => YOUR CHANGES LIKE HERE!

  AT ANY POINT, YOU CAN "RESET" THE STATE BY DISCARDING THE CONTAINER INSTANCE => ./run-container --clean
Bob's Router

You can't connect to Bob's router directly - do CSRF on Bob instead!

1. Run arbitrary JS on Bob's browser (CSRF attack)
   → Starting point: Fetch main page of Bob's router (http://router.local)

2. Learn about exploit you can run on router to run arbitrary PHP code → Run a "reverse shell"
   → More info in docs! Find

3. Poke around Bob's router to flag!
TOOLS THAT MIGHT BE HELPFUL FOR Bob/ROUTER

1. RECEIVE HTTPS REQUESTS SO YOU CAN SEE CONTENTS
   ⇒ NETCAT EXAMPLE FROM XSS LECTURE (LECTURE 10)

2. GET Bob/ROUTER/USER TO LOAD YOUR WEBPAGE
   ⇒ USE "SIMPLE WEBSERVER" TO HOST FILES LOCALLY ON YOUR SYSTEM

⇒ LOOK FOR AN ANNOUNCEMENT w/ MORE INFO SOON!

⇒ LOOK FOR AN ANNOUNCEMENT w/ MORE INFO!