Flag Gearup

Goals

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

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

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.

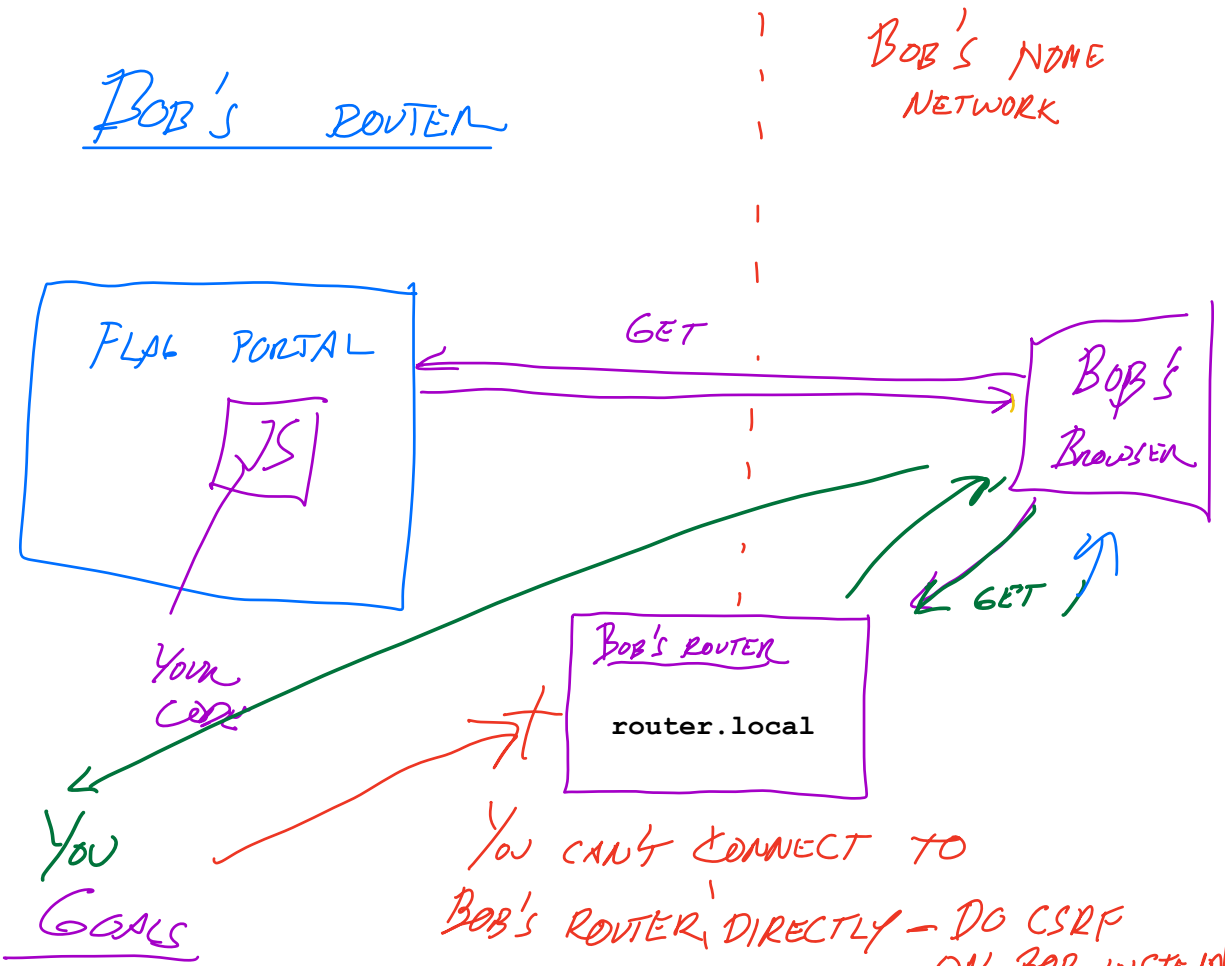
- Container instance ("container"): created when container started, read-write

=> created when you run ./run-container
./cs1660-run-docker

Your changes live here!

At any point you can "reset" the state by discarding the container instance => `./run-container --clean`

Demo: Container setup



GOALS

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 EXPLOITS YOU CAN RUN ON ROUTER TO RUN ARBITRARY PHP CODE ⇒ RUN A "REVERSE SHELL"
   - ↖ MORE INFO IN DOCS/ FIND CLASS SOON!

3. POKE AROUND BOB'S ROUTER TO FLAG!

TOOLS THAT MIGHT BE HELPFUL FOR BOB'S ROUTER

1. RECEIVE HTTP REQUEST 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!