Gearup

Project 2: Flag

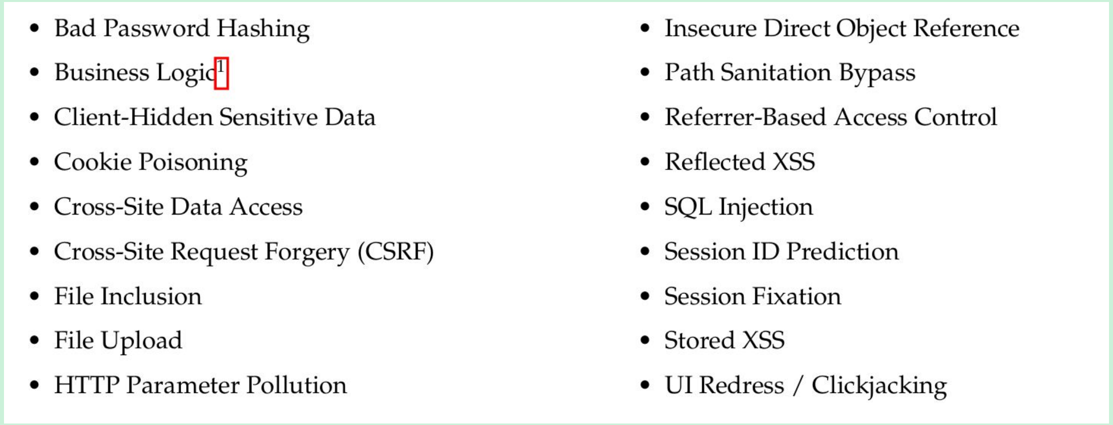
Goals

Learn about web security by attacking a broken website.

- Poke around the site, understand how the system works and 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** five (5) vulnerabilities
- Each must be from a **distinct** vulnerability category



- 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

The Wiki

We’ve provided a wiki that explains each vulnerability in detail:

https://brown-csci1660.github.io/flag-wiki/

Use the wiki to:

- Learn about each type of attack and how it works
- Understand the “Criteria for Demonstration” => what you need to show us

Getting Started

- "Flag portal container": Download a container on your system => Hosts the website for you to attack
- Use (almost) any tools on your computer to find vulnerabilities and break the system:
  - "Developer tools" in your browser (Firefox <u>highly</u> recommended)
  - Your dev container from Project 1 (for Linux tools, running scripts, etc.)
  - Burp Suite
  - Anything else so long as it doesn't <u>automatically</u> find vulnerabilities for you

Getting Started

Project setup guide:

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

What’s in this guide:

- How to setup the “flag” container
- Helpful resources if things go wrong with the containers

Demo: Container setup

Important container terminology

- Container image (“image”): **read-only** package with files/settings for how the container runs.
- Container instance (“container”): created when the container runs, and is **read-write**

Your changes live here!

To reset back to the original container state:

```
> ./run-container --clean
```

Submitting

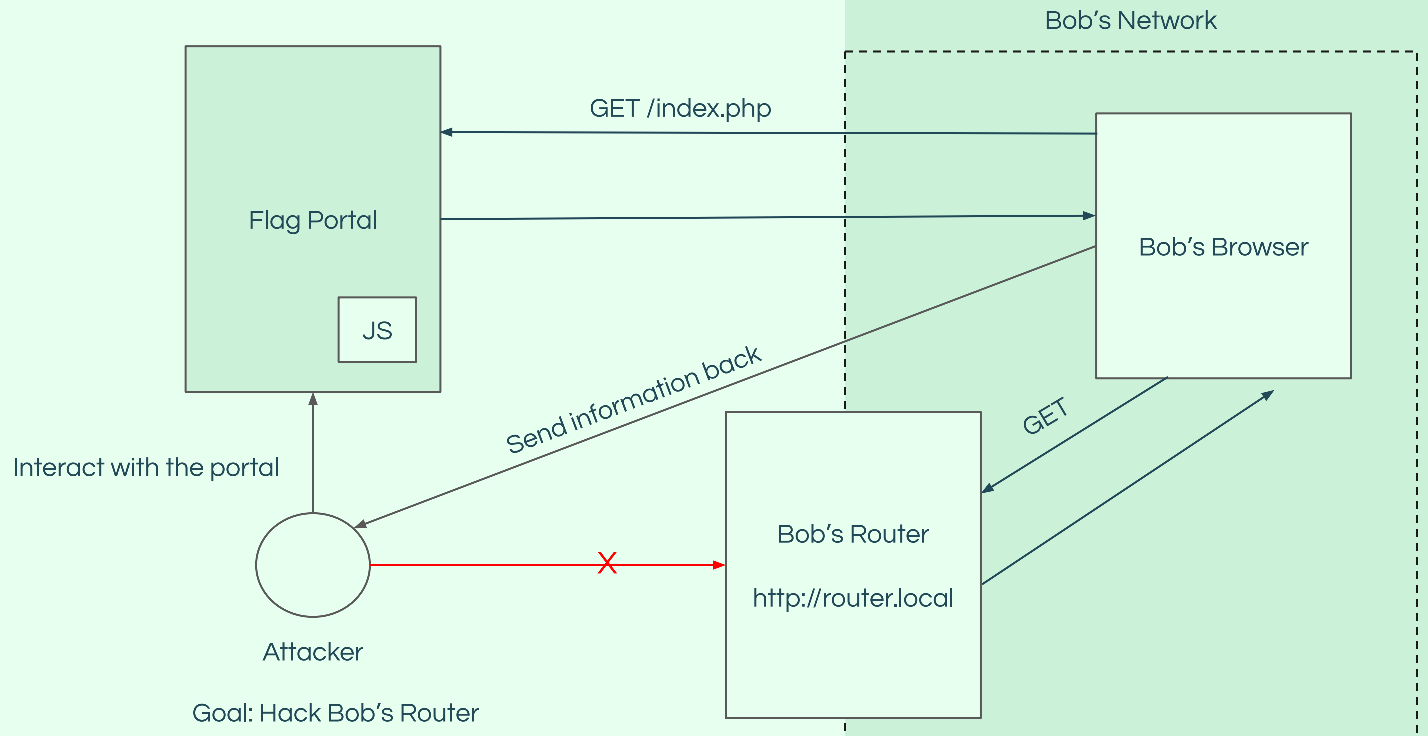
To submit your work, you should commit your:

- README with vulnerability reports (as single PDF)
- Demo video (.mp4 or Drive link)
- Payloads, code or any files that are needed to carry out your exploits

Upload it to the “Flag” assignment on Gradescope

Bob’s Router (CS1620/2660 only)

Overview


Bob's Network
Flag Portal
JS
GET /index.php
Bob's Browser
Send information back
GET
Interact with the portal
Bob's Router
http://router.local
Attacker
Goal: Hack Bob's Router

Goals

- Run arbitrary JS on Bob’s browser (CSRF attack) => try to fetch the main page (http://router.local)
- [Spoiler] The main page will be a login page => based on what you know about the router, try to log in!
- Based on the content of the router’s page, you’ll learn about an exploit you can run on the router to run arbitrary PHP code => try to run a “reverse” shell
- Poke around the filesystem until you get the flag!

Tools & Resources

- Lectures 10-12 => your starting point to web security!
- Simple webserver => a tiny container to host arbitrary pages
  - Use this if you need to have your target download something.
- webhook.site => a temporary URL that logs any requests
  - Can use as a target for CSRF attacks, etc.
- PHP reverse shell
- Tools that may be helpful for Bob’s router:
  - For CSRF attack => Send page contents to webhook.site, or netcat example
  - Get Bob/router/user to load your webpage => Use “simple webserver” to host file locally on your system