Introduction to Web Exploitation

Andrew Haberlandt

January 19, 2021

- **Don't forget to start recording**
- Slides are on https://wiki.osucyber.club

Opportunities this week

- Women in Cybersecurity Meeting, Thursday @ 7pm
  - <insert topic for this week>
  - <insert website/meeting link / mailing list>
- Cyber Security Club Bootcamp CTF continues...

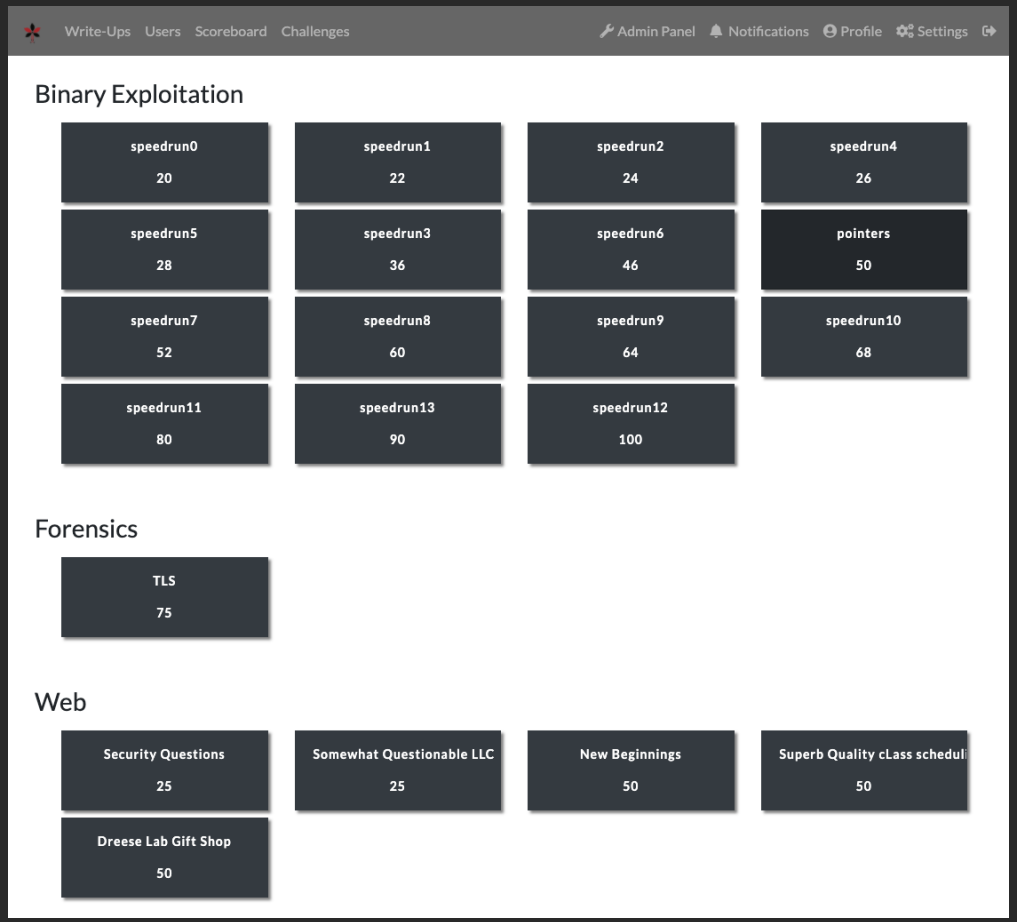
Overview

What is Capture-The-Flag

- **Essentially a hacking competition:** Each challenge contains a 'flag', which is just a secret string. For example: `osuctf{th15_15_n0t_a_r34l_flag}`
- **All of our flags are formatted like** `osuctf{...}`
  - **Sometimes, we'll give you a file:** You'll might have to decrypt some data, reverse engineer an executable, or analyze network traffic
  - **Sometimes, we'll give you an IP and port:** We have set up over 20 internet-accessible services for you to hack: these include websites and APIs, as well as embedded device applications (eg. like on IoT devices)

"CTF Scoreboard"

- Each challenge has an assigned point value, depending on difficulty
- Solve in any order



Categories (our definitions)

- **Web:** Vulnerabilities in web applications such as SQL injection, command injection, cross-site scripting (XSS), logic bugs, and more.
- **Reversing:** Figure out how a program works, without source code. (includes mobile applications, web applications, IoT firmware, ...).
- **Binary Exploitation:** Identifying and exploiting memory corruption and logic bugs in native executable programs. This includes some analysis of machine code and often C source code.
- **Crypto:** Learn about implementation flaws in encryption schemes that allow you to decrypt encrypted data sent between two parties
- **Forensics:** Recovering useful information from traffic captures, full disk images, a variety of common file formats (including data hidden in images). Often includes deleted / covertly recorded data.

Why should you care?

- **You want to write secure software.** Someday, you'll probably write the same kind of applications we've set up for you to hack. You'll be able to recognize a variety of vulnerabiities and avoid introducing them in your code.
- **Breaking other people's stuff can be fun and profitable.** Most large companies have started *bug bounty* programs, which pay anywhere from $100 - $1,000,000 for vulnerabilites that impact the security of their systems and users.
  - HackerOne and Bugcrowd are two common platforms for corporate bug bounties. Make sure you follow all the rules for any programs you participate in.
- **Cybersecurity is a hot CS career field.** The skills you learn here are highly valueble to big tech, big defense, big finance, etc.

About Cyber Security Club Bootcamp

- **Bootcamp CTF:** Runs all semester. Complete challenges online.
- **Bootcamp Series Talks:** Will continue for... unknown period of time
  - During Cyber Security Club meetings, Tuesdays @ 7pm, http://zoom.osucyber.club
  - Talks are recorded and posted on the meeting schedule on https://wiki.osucyber.club
  - Each talk will conclude with a list of recommended challenges
  - **Next Week:** Intro to Cryptography
  - Most talks are standalone, exceptions will be announced and listed on wiki
  - Some topics will (eventually) have a 'Part 2', Web Exploitation is one of them
- **How to get Started: https://go.osu.edu/CyberClubBootcamp**

Rules

1. **Don't spoil the challenges for others.**
   - You are free to discuss approaches to the challenges, tools you used, important concepts, and small hints on our Discord
   - Please do not give flags or solution guides/scripts to other participants.
2. **Don't try to hack the CTF scoreboard (CTFd) or the Discord.**
3. **Be nice on the Discord.**

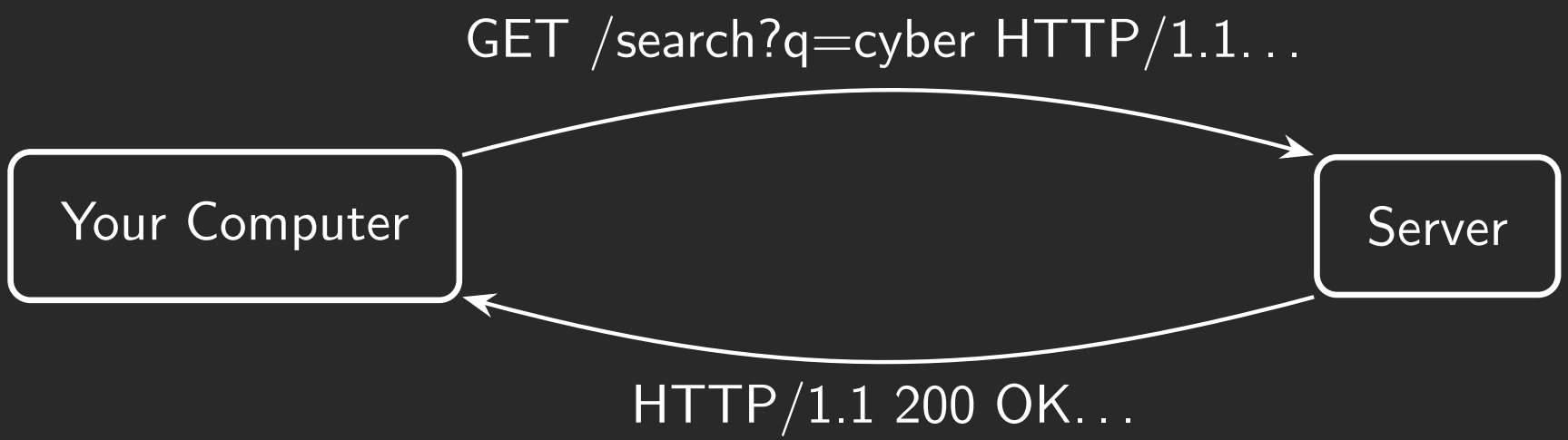
Prizes

- Prizes (while supplies last)
  - 500pts: Sticker (100 available)
  - 2000pts: T-shirt (30 available)
- Pick up on-campus starting in Feb, or if you are remote you can wait until AU21

About the web: HTTP

- You want to go to http://google.com . . . How?

1. DNS lookup for google.com → IP address
2. Open a TCP connection, send an HTTP request message
3. Parse HTTP response message
4. Parse and render the HTML in the response
   - Sometimes this requires additional requests for external resources, which goes back to #1



Why do we care?

- There are lots of websites, with lots of data
- APIs
- Web technologies bleeding into desktop apps (Electron)
- Bug bounties are largely web applications (e.g. HackerOne, Bugcrowd)

What *is the 'Web' Category?*

Types of vulnerabilities to consider

- **Sensitive data exposure / information leakage:** Can you get the server to give you information you shouldn't have access to?
- **Broken Access Control:** Can you modify data on the server without proper authorization?
- **Broken Authentication:** Can you login as another user or compromise their password or session tokens?
- **Manipulating Responses to other users:** Can you modify resources provided to other users in a way that will directly or indirectly give you access to their account?

Ethics

- **Get explicit permission from the vendor, follow all rules they give you**
  - Bug bounties are generally very specific about scope
- **If you break into a system for which you are not authorized, you will probably get caught.**
  - Advice: Don't touch anything at the university. "Use only those computing resources they are authorized to use and use them only in the manner and to the extent authorized" (University Policy) - they will find you and come after you
  - Computer Fraud and Abuse Act - many years in prison
  - Won't be able to get a security clearance
- **Responsible Disclosure:** If you find a security bug, you should report it to the vendor and give them time to fix it before disclosing publicly
  - Exception: Bug bounties that make you sign an NDA – you can never disclose
  - Exception: Most researchers give the vendor 90 days to fix (see Google Project Zero), and then publicly disclose even if not fixed

Demo: Chrome Developer Tools

Tools

- Chrome Developer Tools
- Postman, Burpsuite, other HTTP clients
- Python 'requests' library
- **Man-in-the-middle proxies:** mitmproxy, Charles proxy
- **SQL injection tools:** sqlmap
- dirbuster, and other **wordlist-based scanning tools**
- w3af, and other **automated vulnerability scanners**

Recommended Challenges

- **25 pt Web:** Security Questions
- **35 pt Web:** scrape
- **50 pt Web:** Auth Bot 2
- **60 pt Web:** New Beginnings
- **70 pt Web:** Dreese Lab Gift Shop
- **75 pt Web:** flagbin 1
  - Hint: Maybe they were worried about SEO?
- **Challenge: 150 pt Web**: Mini Calculator
- SQL Injection Challenges
  - **25 pt Web:** Somewhat Questionable LLC
  - **50 pt Web:** Superb Quality cLass scheduling
- Also, any challenge 25 pt or less should be solvable before we talk about that category