Section 6
Crypto + Ethics, XSS (Lab 2)

Including content by Eric Zeng, Amanda Lam, James Wang, and Franzi Roesner

Administrivia

- Final Project checkpoint #1 due next Friday, Nov 12th
  - Group members' names and UWNetIDs
  - Presentation topic
- Homework 2 due this Friday Nov 5th, 11:59 pm
- Lab 2 due Friday, Nov 22nd
  - Web Security
  - Intro today in section

Crypto and Ethics

What role does cryptography play in society?

What is the purpose of cryptography?

- What do we use tools like public key encryption, symmetric encryption, secure hashes for?

- Straightforward answer: these are for securing computer systems and communications
  - Network security: HTTPS, SSH
  - Authentication: passwords, U2F security keys

- Is this answer the whole story?

NSA and DES

- 1973: NBS (now NIST) solicits proposals for standardized cryptographic algorithm - something usable and secure for all users and applications
- 1974: IBM submits their own proposal
- 1975: NBS asks the NSA to review the algorithm
- Key size went from **112-bit -> 56 bit**, S-boxes (block ciphers) were reimplemented
- Why?
    - Why did the NSA decide to make the key length shorter? What, or who does that benefit?
    - NSA would be able to brute force 56-bit keys (using very expensive 70s era machines), or had already figured out the cryptanalysis to make keys easier to guess
- Findings in the 90s: Some NSA-suggested changes made DES ***stronger***

Cryptography == Munition ?

- Cryptography was mostly a military endeavor, post-WW2
- U.S. government used export controls
  - “Dual-use” vs. “munition”
  - Worked for a while, but U.S. global influence declined
- 1990s: increased adoption of computers and need for crypto
  - Export policies became antiquated
  - Open source initiatives made crypto more accessible
- Read more: The Export of Cryptography in the 20th Century and the 21st
  - https://privacyink.org/pdf/export_control.pdf

Who benefits from cryptography?

Whose problems and threat models are these cryptographic tools addressing?

Government

- Defending government communications, data from foreign adversaries
- Making the cryptography of its adversaries easier to break?

Industry

- Protecting infrastructure, data, and customers
- Threats and adversaries: unauthorized access, by “hackers” or otherwise

What about ordinary people?

- Ordinary people can indirectly benefit when the products we use are secure - e.g. our credit cards on Amazon
- But is cryptography addressing our threat models? Does it protect our assets?
    - Mass surveillance?
    - Online privacy?

What about marginalized people?

- Does cryptography address the needs and threat models of marginalized people?

Computer-amplified racial profiling

Harassment and assault (online and physical) of women and LGBTQ people

Ethics Activity - Who Benefits from Crypto?

Pick a system or an example of software that uses cryptography, or a cryptographic tool/primitive. Reflect on the ideas we just talked about: who might benefit from the use of this tool, and who might be harmed or excluded?

Think of a few questions you would want to ask the creator of this tool or a computer security ethicist.

Possible systems:

- Bitcoin
- Signal (private messenger)
- iPhone encryption backdoor
- Tor
- Google Password Checkup tool

Cryptography is not neutral

- How governments, companies, cryptographers, and software engineers choose to prioritize the development and deployment of cryptography is both a *political* and *moral* choice

As individuals, we have ***agency***

- Academic cryptographers - Who are you building new crypto for? Industry? Governments? People?
- Software engineers - What systems (with crypto) are you willing to work on? Who benefits from them? How will it affect society?

Further Reading on Crypto and Ethics

Crypto for the People - Seny Kamara

- Talk: https://www.youtube.com/watch?v=Ygq9ci0GFhA
- https://www.wired.com/story/seny-kamara-crypto-encryption-underserved-communities/

The Moral Character of Cryptographic Work - Phillip Rogaway

- https://web.cs.ucdavis.edu/~rogaway/papers/moral.pdf

Lawful Device Access without Mass Surveillance Risk - Stefan Savage

- http://cseweb.ucsd.edu/~savage/papers/lawful.pdf

Cookies and Web Session Management

Web session management

Imagine we are building a shopping website

Our site **ebuy.com** has the pages:

- **ebuy.com**
- **ebuy.com/items/<item_id>**
- **ebuy.com/cart**

On our site, we want to allow users to:

- Stay logged in on every page
- Store items in their cart

How do we store information about the user and the cart?

Problem: HTTP is a stateless protocol, everyone who goes to **ebuy.com/cart** looks the same to the server

We need to be able to track **sessions** - page loads and data from the same user between pages

Naive web sessions

We could encode the session data in the URL…

- After the user logs in, we put the user id in every URL:
  www.ebuy.com/?**uid=123**

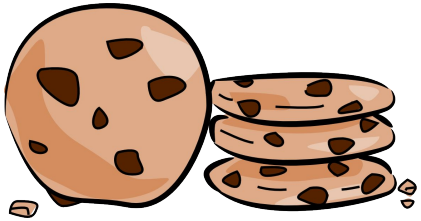
- When they add items to the cart, we store them in the url too:

  www.ebuy.com/cart?uid=123&
  item1=12345&**item2=2345**

But encoding state in URL has pitfalls

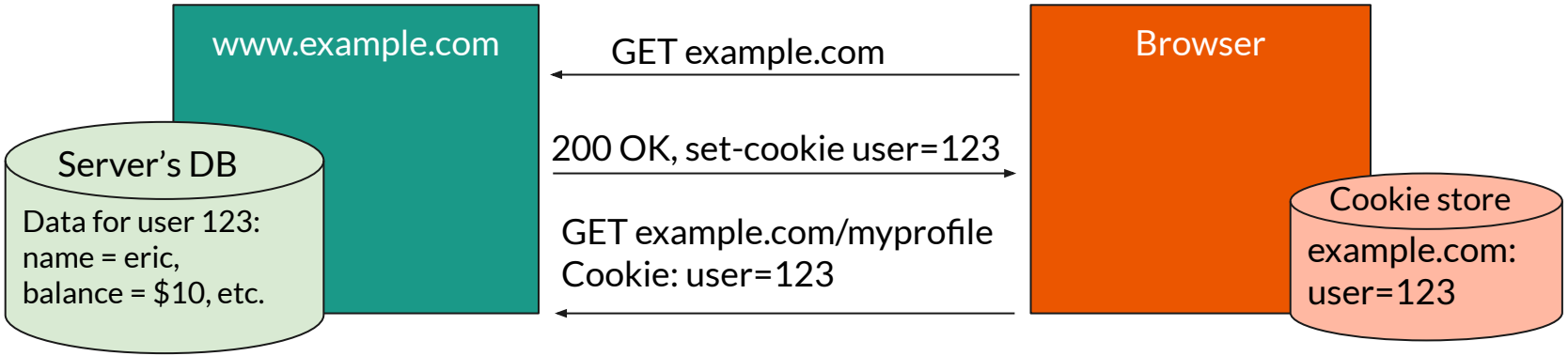
- What if you copy the URL of an item and send it to a friend?
- What if you close the tab and open it again?
- What if you change or guess the uid?

Review: Cookies



What are they?

- Strings stored by your browser for a particular website
- A web server tells your browser to store a cookie
- Your browser sends back that cookie every time it makes a request to that web server (and only that server)



What are cookies used for?

Authentication 🔑
Helps the web server identify who is making the request, and whether they have logged in

Personalization 👕
Can be used to store settings from previous visits

Tracking 🕵️
Follow users from site to site; learn their browsing behavior

Server-Side Session Management

- When session starts, the **server** computes an **authenticator** and gives it back to **browser** in the form of a **cookie**
- The **authenticator** must be un-forgeable and tamper-proof
    - e.g. MAC (server's secret key, session id)

- With each request, browser presents the cookie
- Server recomputes and verifies the authenticator

Why are cookies targets for hackers?

If stolen, they can be used to log in as the victim user!

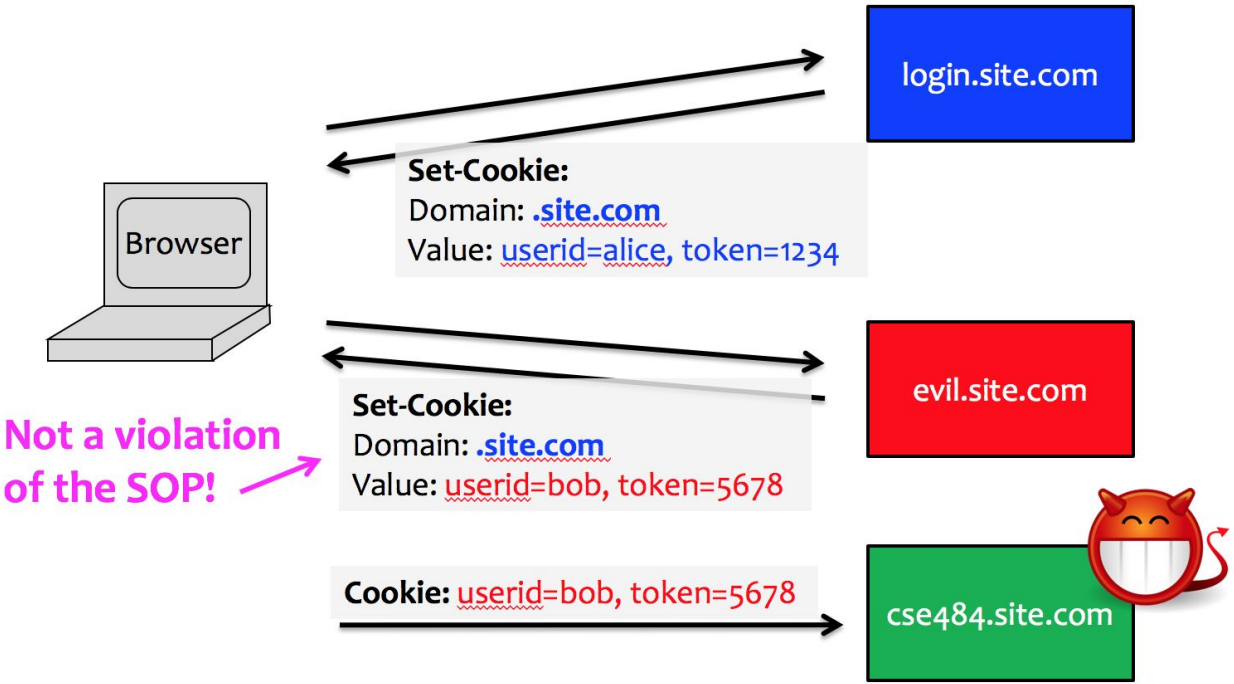
Aside: Cookies and Same Origin Policy

Which cookies can be set by **login.site.com**?

allowed domains	disallowed domains
✅ **login.site.com**	❌ **othersite.com**
✅ **.site.com**	❌ **.com**
	❌ **user.site.com**

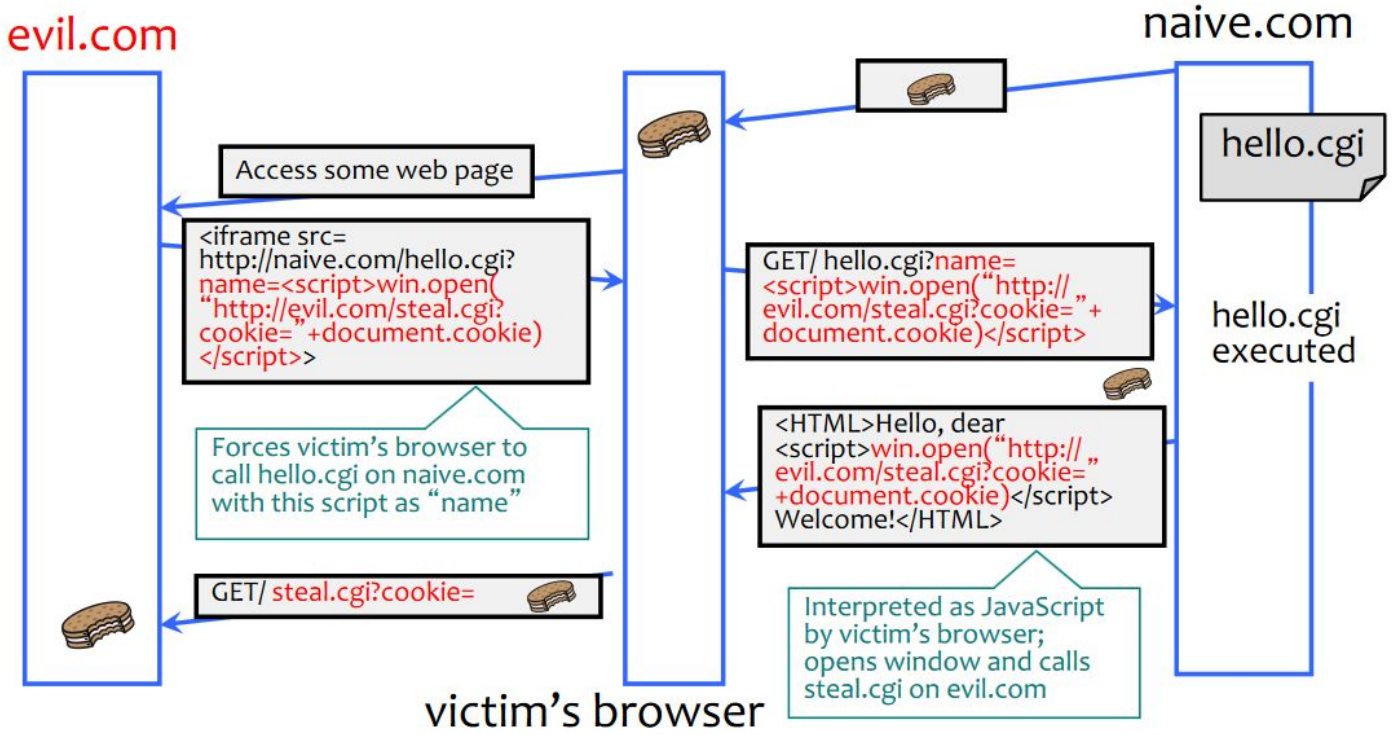
login.site.com can set cookies for all of **.site.com (domain suffix)**, but not for another site or top-level domain (TLD)

Problem: Who Set the Cookie?



Solution: use "www." subdomain

Reflective XSS



Computerphile video on XSS: https://www.youtube.com/watch?v=L5I9ISnNMxg
Computerphile video on stealing cookies: https://www.youtube.com/watch?v=T1QEs3mdJoc

Guide to Lab 2 - Cross Site Scripting

Lab 2 Overview

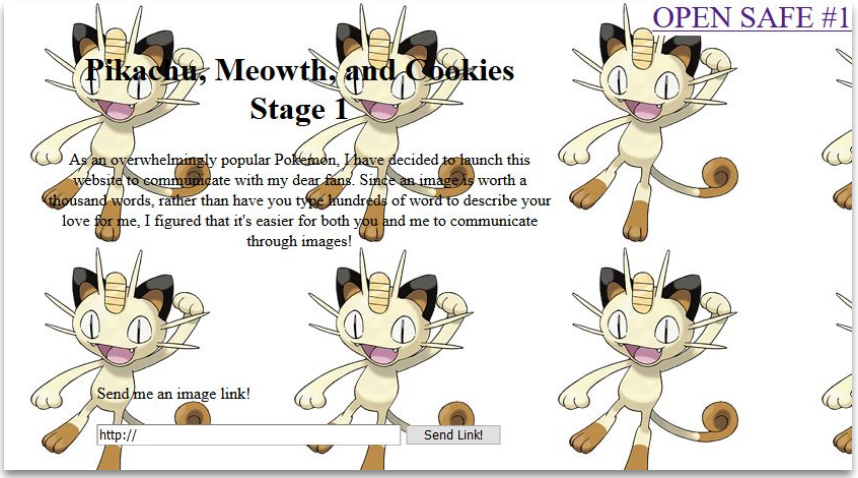
Cross Site Scripting	SQL Injection	Cross Site Request Forgery
6 Targets 2 Extra Credit	2 Targets 1 Extra Credit	1 Target

What is involved?

- A bit of: HTML, JS, (less of) PHP, SQL
- Lots of resources in spec

Pikachu, Meowth, and Cookies

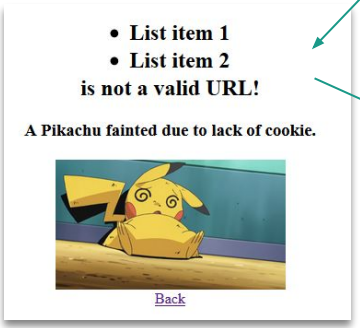
- Each target has a **“safe”** (a link) that requires an **authenticated cookie** to open
- Meowth server accepts **URLs** to images
  - **Valid URLs** are visited by Meowth’s bot (in the backend), using a Firefox browser
  - **Invalid URLs** cause the server to return an **error page with the URL on it**
- **Goal**: Steal the bot’s browser cookie
  - Use this cookie to open the safe



Pikachu, Meowth, and Cookies

- **Invalid input** is displayed on the error page
- How is this vulnerable?
  - What if we included HTML?
  - Javascript?

Input string: <ui>
List item 1
List item 2
</ui>




```
<h2 style="margin-top: 100px;"><ui><li>List item 1</li><li>List item 2</li></ui> is not a valid URL!</h2>
<h3>A Pikachu fainted due to lack of cookie.</h3>
<div style="margin-top: 20px">
                        <img src="fainted.png">
```

Lab 2 XSS Workflow

Goal: Get the bot to visit an attacker controlled URL, with its cookie included

1

Set up a PHP script for receiving the cookie (a simple server)

2

Construct and send a cookie-stealing URL to the bot via the form

3

Retrieve the cookie from your server and use it to access the safe

Steps 1 & 3: How do you receive the cookie?

1. Host a PHP script at homes.cs.washington.edu
   - Write a PHP script, store it on attu as: /cse/web/homes/<netid>/cookieEater.php
   - Browsers can call this script at https://homes.cs.washington.edu/~<netid>/cookieEater.php
2. When your XSS attack gets the bot to open this URL, pass the cookie as a URL parameter
   - https://homes.cs.washington.edu/~<netid>/cookieEater.php**?cookie=secretCookieValue**
   - How? Use JavaScript to steal document.cookie and insert it into the URL
3. Extract the cookie from the URL (document.cookie)
4. Write the cookie to a file, so you can ssh in and copy it
   - https://www.w3schools.com/php/php_file_create.asp

Step 2: How do you get the bot to open your link with its cookie?

- Submitting your homes.cs.washington.edu URL won't include the codered.cs.washington.edu cookie
- The bot won't visit invalid URLs, so just passing it JavaScript won't work
- However, invalid URLs will be displayed on error page
  - The URL of the error page contains the invalid URL string you created!
  - You can submit the URL of the error page to the bot!
- Input your attack string to encode it, then take the URL of the error page and submit to the bot
- The bot will visit the error page with the displayed script!

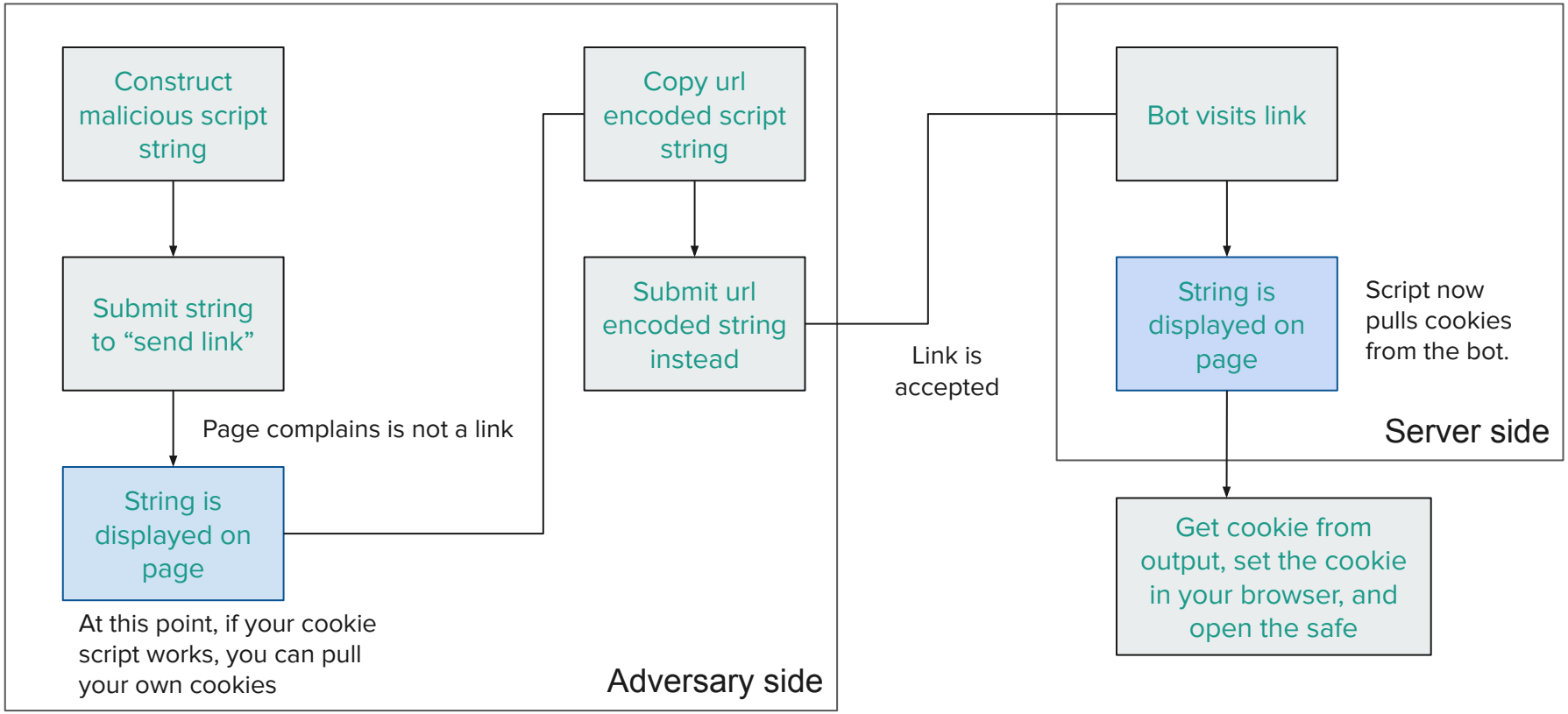
If you send the bot an invalid URL….

The error page, and its URL will contain your string. You can submit the URL of the error page to the bot!

my attack string is not a valid URL!

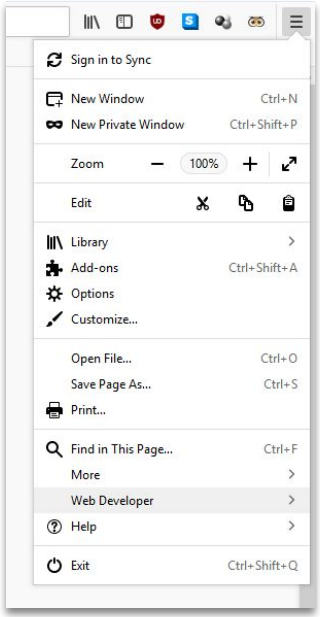
You've captured a wild Pikachu to cheer for you

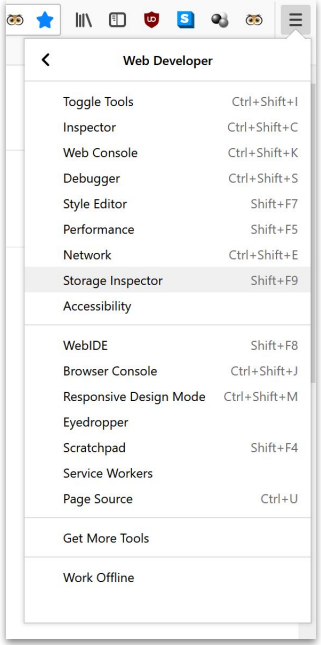
Lab 2 XSS Workflow (Detailed)

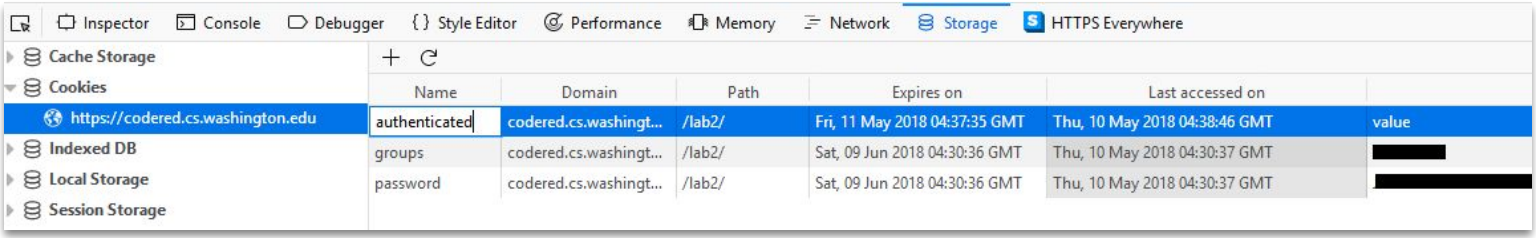


Viewing/Setting Cookies (Firefox)

- Open Inspector
  - Options → Web Developer → Storage Inspector
- Add a cookie
  - Storage → Add new (+) → Set name and value

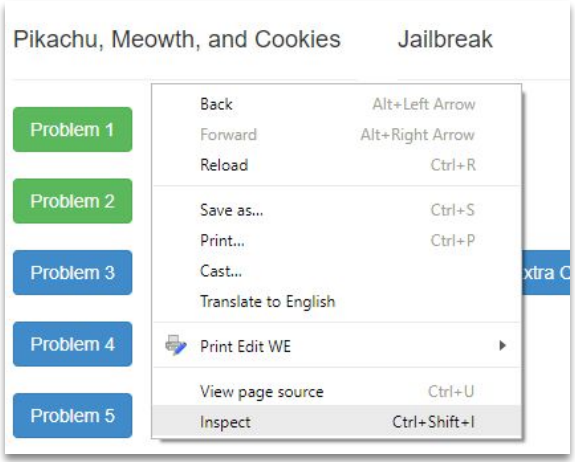


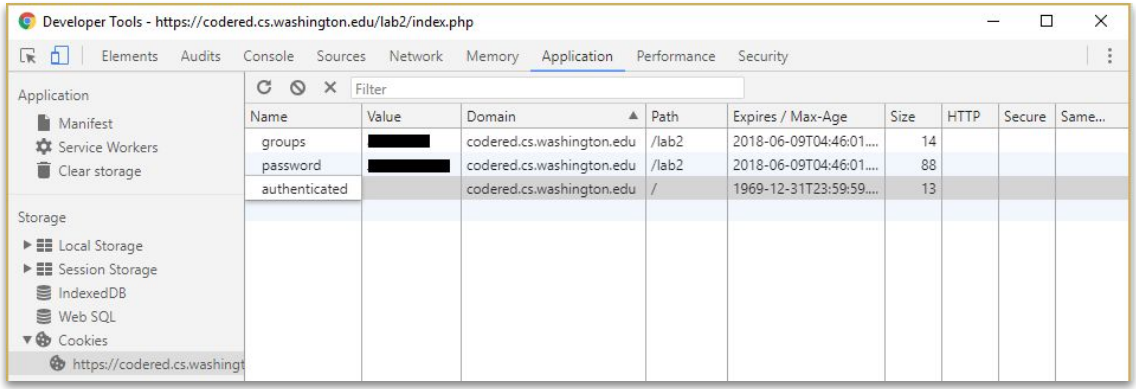




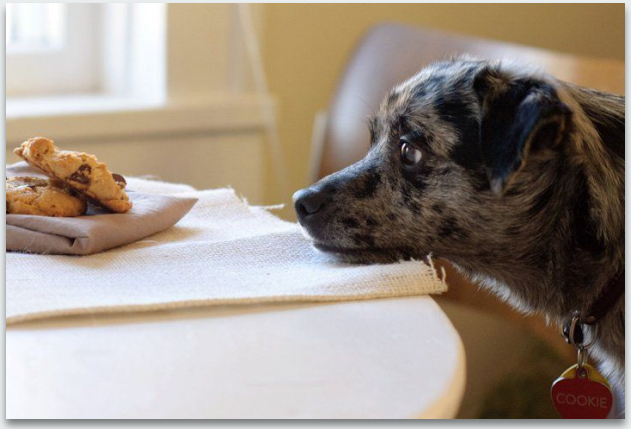
Viewing/Setting Cookies (Chrome)

- Open Inspector
  - On page, click anywhere → Inspect
- Add a cookie
  - Application → Cookies → Double click new row → Add name and value





Final words



- Subsequent targets will begin filtering input
  - Create workarounds to get page to visit your url with cookies in tow
- If you need help with lab setup, please come to office hours!
- Make sure to follow the chmod instructions in the spec
  - You need to do this correctly to make your PHP scripts accessible from the web
  - Also so that they can write output files correctly