masoncc.slack.com

Visit this or i h8 u

Why are sites hackable?

- Unaudited code
  - You don’t actually check the code of your WordPress install
- Shit tier web developers
  - Understand how to code, but not how to code securely

Different Website Langauges

- PHP
  - Where 99% of your web exploitation will take place
- ASP.NET (.NET framework)
- Ruby (Ruby on Rails, Sinatra)
- Django
  - SRCT uses this

And others

Cross Site Scripting (XSS)

What is it? Allows hackers to inject Javascript (or potentially another similar language) into a page

Impact? Allows people to impersonate users, cause unwanted behavior, execute browser-side Javascript

Remediation? Don’t take input from a URL or a database and just immediately display it to a user. Sanitize it, filter it, make sure it’s expected input, escape (change the characters) it.

Different Kinds of XSS

Reflective:

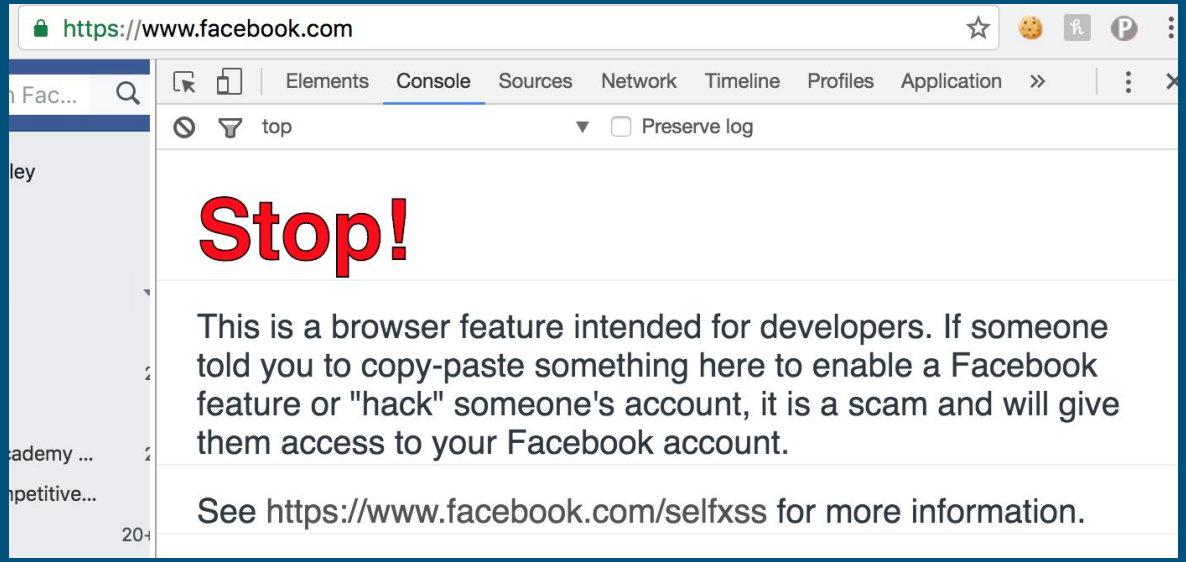
- Chrome and the like try to stop this actively
- Tl;dr usually the kind that has Javascript in the browser URL
- Users who inherently trust the website can get owned by clicking the URL

Stored:

- Way nastier, browsers don’t generally screw with it
- When unwanted code is expected to be text and pulled from the database/server side and displayed to the user, executing the code

Self XSS

User is social engineered (tricked) into going to the developer console of a browser and executing malicious code



SQL Injection

What is it? Data accepted from a user is put directly in a database query/statement that allows the user to interactively change the syntax of the statement, allowing the user to

Impact? Huge. Database leakage, information leaked (which is why password storage is also so important).

Remediation? Depends on language, but bobby-tables.com is great for this. In PHP, use prepared statements, restrict the character set to UTF8, filter/sanitize user input.

SQLi Expanded

- **Go to go.gmu.edu/sqli** for a few more examples on SQL injection
- Note: Securely store passwords. LinkedIn and other websites have gotten their databases hacked and leaked, and if their passwords were securely “hashed” they wouldn’t be nearly as screwed. If you hear “MD5”, “SHA1”, etc they are not good for storing passwords.

Command Injection

- Like XSS on the server side?
- Accepts user input, doesn't filter/sanitize, passes it to the system to execute
- Results in the user being able to run commands on the server
  - ***Really bad***
- Code injection is also a thing, where PHP and other languages can be executed
- This is where knowing the system comes in handy (i.e. how to use Linux)
- Go to **go.gmu.edu/cmdi** for more examples

CSRF - Cross Site Request Forgery

- Todd runs website A (banking website) and website A initiates a transfer when a logged in user visits https://hi.com/index.php?transfer=500&to=Michael
- Michael running website B embeds an image or content and claims the source URL is “https://hi.com/index.php?transfer=500&to=Michael”

CSRF LIVE EXAMPLE (SORRY SRCT)

Go.gmu.edu recently had a CSRF ticket filed on Gitlab

- We own go.gmu.edu/mailinglist for our mailing list
- I log into Go (or any similarly GMU-authenticated site really) and visit michaelbailey.co
- Michaelbailey.co has an image embed of go.gmu.edu/delete/mailinglist
- Our URL gets delisted and deleted

Examples

Pico CTF 2013 GETKEY

Hack The Vote CTF

Hack This Site Walkthrough

Hackthissite.org

Walkthrough at go.gmu.edu/hackthissite