RAILS SPRING 2014

week 9

AGENDA

- Security
- Authorization w / Can Can
- Factory Girl for test data

SECURITY

- Least Privilege
- Secure Defaults
- Defense in Depth

least privilege
(deploy user)
secure by default
- whitelist
- safe api - active record query api - safe way (escaping)
automatic sanitizing (cross site scripting default)

Defense in Depth
- layers of defense - layers buy more time (security)
- like not worry about script injection in Admin area -> whats the impact?

SECURITY ISSUES

- Session Hijacking (firesheep)
- Brute Force Attacks (psswd stolen, sha1: 9 chars, 7 years)
- Insecure Direct Reference (proper auth)
- Mass Assignment (strong_parameters)
- CSRF (watch for GET)
- Offsite Redirects
- Unexpected Code Execution (routes, and render)
- SQL Injection (http://rails-sqli.org/)
- XSS (sanitize)
- YAML Injection (http://blog.codeclimate.com/blog/2013/01/10/rails-remote-code-execution-vulnerability-explained/)

Attacks:
1. session hijacking - firesheep (made it easy, drew attention to it, ssl for login but not for every view
- force ssl (everywhere) -> config.force_ssl = true (nginx for ssl termination)
- secure and httponly cookies (done for you on most rails versions - should have secure and httpOnly)
- strict transport security - force ssl sends this by default
- require a password for sensitive stuff
2. craft the url to send params -> make sure you are checking proper authorization for data you are presenting (roles? maybe over-kill…) -> force checking authorization
3. mass assignment - rails 4 has strong_parameters - requires a object (hash) in the params then permit what fields are allowed in that hash
4. csrf - only apply to post etc.. so you have to be safe with GETs
5. leaky routes - not a default thing - you make it accessible -> strict limits on actions in routes, also user specifications in render
Offsite redirects
Good: Verify protocol and host
Better: Create a whitelist
Best: Use an identifier instead of a URL
6. sql injection (use the query interface) but be careful know the AR api - uncommon (calculate and pluck)
rails-sqli.org
7. cross site scripting (XSS) -> malicious js -> Rails 3+ -> rails_xss, sanitize user generated html (Loofah gem) -> content security policy (tell the browser what is safe) -> raw and html_safe and link_to can inject values if u generate html from user settings ….

BRAKEMAN

- http://brakemanscanner.org/
- https://github.com/presidentbeef/brakeman
- http://brakemanscanner.org/docs/options/

BRAKEMAN

- gem install brakeman
- brakeman -f html > brakeman.html

AUTHORIZATION

- Can Can Can!
- Check abilities to users
- Works nicely with Rolify

FACTORY GIRL

- Getting Test data into your tests
- Factories for different types of models
- Sequences for fake data
- Associated factories for relationships

HOMEWORK

- Prepare to present your project
  - What 3 things would you have done differently with your project?
  - What 3 things do you want to add to your project?
  - What 3 things do you wish you knew when you started?
  - What 3 things did you learn that I didn't go over in class?