Web Security

- The tangled Web
- Lab 5

What you need to know

- Same Origin Policy (SOP)
- Cookies
- Common Exploits & how to fix them

Same Origin Policy

http:// www.csail.mit.edu :80

(Protocol , Host , Port*)

* Except for IE, of course … IE doesn't care about ports

Same Origin Policy (2)

- Allow only same-origin access to resources

  Exceptions:

  - Images
  - Scripts
  - CSS
  - Plugin content

Same Origin Policy (3)

- “Special” origins
  - about:blank
  - about:config
  - javascript: URLs
  - data: URLs
  - blob: URLs
  - etc.

Cookies

- Mantain State
- Managed per domain, i.e. www.mit.edu
- www.mit.edu can set cookie for *.mit.edu!
- Some special flags: “secure”, “httponly”

Basic Exploits

- XSS (Cross-Site Scripting)
- CSRF (Cross-Site Request Forgery)
- Session Hijacking
- Clickjacking

XSS (Cross-Site Scripting)

- Exploits failure to sanitize user content

  Example:

  http://foo.bar.com/books?author=J.P.+Sartre

  could also be:

  http://foo.bar.com/books?author=<script>alert(1)</script>

- Solution: Always escape user-provided input

CSRF (Cross-Site Request Forgery)

- Attacks insecure forms on websites
- Scenario:
  - Bob has a session cookie from alice.com
  - alice.com has a CSRF vulnerability
  - Eve tricks bob into visiting evil.com
  - Javascript code of evil.com can submit to alice.com with Bob's credentials.
- Solution: Use CSRF tokens for every form

Session Hijacking

- Sniffing packets to get session cookies
- Session fixation (setting user's cookie)

SQL injection

- Run arbitrary SQL queries

  Example:

  foo.bar.com/books?author=' AND 0 UNION SELECT …

- Solutions:
  - escape user input
  - Use a library to construct SQL queries

Clickjacking

- Trick users into clicking things they don't want
- Overlay transparent frames

- Solutions:
  - X-Frame-Options: deny
  - Non-embeddable confirmation page