Common Web Security Holes

Paul Grayson
2012-12-05

Some problems with the web

1. You can't trust the network
2. You can't trust URLs and HTML
3. You can't trust user-submitted content
4. You can't trust sessions
5. You can't trust browsers

Introduce demo site

1. You can't trust the network

- Requests and pages can be intercepted
- “Man-in-the-middle”
- DNS hijacking

A partial solution: SSL ($$)

2. You can't trust URLs and HTML

- Path traversal
- Form hacking
- Cross-site request forgery (CSRF)
  - Rails: authenticity-token blocks this.
  - That makes it hard to do other stuff, like caching or emailing links that take an action.
- Cross-site scripting (XSS)
  - Chrome blocks scripts in parameters

3. You can't trust user content

- Persistent XSS
- SQL injection

In Rails: HTML and SQL strings are cleaned (quoted) automatically in most cases.

But it is really easy to inadvertently bypass this protection.

4. You can't trust sessions

- Session modification
  - Rails: generally blocked
- Session fixation
  - *e.g.* via XSS attack
  - Rails: reset sessions on login
- Session replay attacks

5. You can't trust browsers

- CSS visited leak
  - Fixed in Chrome and Firefox April 2010
  - Not fixed in IE
- 403 error logged in leak
- “Clickjacking”
- Etc...

Conclusion

- Rails and modern browsers help.
- But there are still a lot of traps!
- Discuss