Keeping Rails Applications on Track with Brakeman

Justin Collins
@presidentbeef
RailsConf 2012
Everyone knows they “should” worry about security
But when should you worry?
Idealized Software Development

Write code
Run tests
Commit code
Push to CI server
Code review
QA Testing
Deploy code
Cost to Fix Defects

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code

Cost to Fix Defects

5
Cost to Fix Defects

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code

Cost to Fix Defects: 6
Cost to Fix Defects

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code

Security Review

Cost to Fix Defects
Cost to Fix Defects

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code

Security Review
Cost to Fix Defects

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code

Security Review

Cost to Fix Defects: 6
Cost to Fix Defects

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code

Security Review
Cost to Fix Defects

Security Review

Write code Run tests Commit code Push to CI server Code review QA Testing Deploy code

Cost to Fix Defects

Security Review

Write code Run tests Commit code Push to CI server Code review QA Testing Deploy code
Cost to Fix Defects

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code

Security Review
Cost to Fix Defects

Security Review

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code

Cost to Fix Defects
Cost to Fix Defects

Security Review

Write code  Run tests  Commit code  Push to CI server  Code review  QA Testing  Deploy code

Cost to Fix Defects

6
Cost to Fix Defects

Security Review

Write code | Run tests | Commit code | Push to CI server | Code review | QA Testing | Deploy code

Save code

Cost to Fix Defects: 6
“zero configuration”
security scanning
gem install brakeman

cd my_rails_app/

brakeman
### SUMMARY

| Controllers | 3 |
| Models      | 2 |
| Templates   | 22 |
| Errors      | 0 |
| Security Warnings | 57 (42) |

<table>
<thead>
<tr>
<th>Warning Type</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Restriction</td>
<td>1</td>
</tr>
<tr>
<td>Command Injection</td>
<td>2</td>
</tr>
<tr>
<td>Cross Site Scripting</td>
<td>20</td>
</tr>
<tr>
<td>Cross-Site Request Forgery</td>
<td>1</td>
</tr>
<tr>
<td>Dangerous Eval</td>
<td>1</td>
</tr>
<tr>
<td>Dangerous Send</td>
<td>2</td>
</tr>
<tr>
<td>Default Routes</td>
<td>1</td>
</tr>
<tr>
<td>Dynamic Render Path</td>
<td>3</td>
</tr>
<tr>
<td>File Access</td>
<td>3</td>
</tr>
<tr>
<td>Format Validation</td>
<td>1</td>
</tr>
<tr>
<td>Mass Assignment</td>
<td>2</td>
</tr>
<tr>
<td>Redirect</td>
<td>1</td>
</tr>
<tr>
<td>Response Splitting</td>
<td>1</td>
</tr>
<tr>
<td>SQL Injection</td>
<td>10</td>
</tr>
<tr>
<td>Session Setting</td>
<td>2</td>
</tr>
</tbody>
</table>

### SECURITY WARNINGS

<table>
<thead>
<tr>
<th>Confidence</th>
<th>Class</th>
<th>Method</th>
<th>Warning Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_command</td>
<td>Command Injection</td>
</tr>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_command</td>
<td>Command Injection</td>
</tr>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_eval</td>
<td>Dangerous Eval</td>
</tr>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_send_first_param</td>
<td>Dangerous Send</td>
</tr>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_another_dynamic_render</td>
<td>Dynamic Render Path</td>
</tr>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_file_access</td>
<td>File Access</td>
</tr>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_file_access</td>
<td>File Access</td>
</tr>
</tbody>
</table>
gem install brakeman

cd my_rails_app/

brakeman -o report.html
<table>
<thead>
<tr>
<th>Configured/Reported</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllers</td>
<td>3</td>
</tr>
<tr>
<td>Models</td>
<td>2</td>
</tr>
<tr>
<td>Templates</td>
<td>22</td>
</tr>
<tr>
<td>Errors</td>
<td>0</td>
</tr>
<tr>
<td>Security Warnings</td>
<td>55 (40)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Warning Type</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Restriction</td>
<td>1</td>
</tr>
<tr>
<td>Command Injection</td>
<td>2</td>
</tr>
<tr>
<td>Cross Site Scripting</td>
<td>26</td>
</tr>
<tr>
<td>Cross-Site Request Forgery</td>
<td>1</td>
</tr>
<tr>
<td>Dangerous Eval</td>
<td>1</td>
</tr>
<tr>
<td>Dangerous Send</td>
<td>2</td>
</tr>
<tr>
<td>Default Routes</td>
<td>1</td>
</tr>
<tr>
<td>Dynamic Render Path</td>
<td>3</td>
</tr>
<tr>
<td>File Access</td>
<td>9</td>
</tr>
<tr>
<td>Format Validation</td>
<td>1</td>
</tr>
<tr>
<td>Mass Assignment</td>
<td>1</td>
</tr>
<tr>
<td>Redirect</td>
<td>1</td>
</tr>
<tr>
<td>Response Splitting</td>
<td>1</td>
</tr>
<tr>
<td>SQL Injection</td>
<td>9</td>
</tr>
<tr>
<td>Session Setting</td>
<td>2</td>
</tr>
</tbody>
</table>

### Security Warnings

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Class</th>
<th>Method</th>
<th>Warning Type</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_command</td>
<td>Command Injection</td>
<td>Possible command injection near line 34: <code>'ls #{params[:file_name]}</code></td>
</tr>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_command</td>
<td>Command Injection</td>
<td>Possible command injection near line 37: <code>system(params[:user_input])</code></td>
</tr>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_eval</td>
<td>Dangerous Eval</td>
<td>User input in eval near line 41: <code>eval(params[:dangerous_input])</code></td>
</tr>
<tr>
<td>High</td>
<td>HomeController</td>
<td>test_send_first_param</td>
<td>Dangerous Send</td>
<td>User controlled method execution near line 81: <code>User.send(params['method'], to_sym)</code></td>
</tr>
<tr>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td>Default route: All public methods in controllers are available as actions in routes.rb near line</td>
</tr>
</tbody>
</table>
Confidence: High

View: home/index (HomeController#test_render)

Warning Type: Cross Site Scripting

Line Number: 5

Code Snippet: Unescaped parameter value near line 5: params[:unsafe_input]
Unescaped parameter value near line 5:
params[::unsafe_input]

/app/views/home/index.html.erb

1  <h1>Home#index</h1>
2  <p>Find me in app/views/home/index.html.erb</p>
3  <%= params[::user_input] %>
5  <%= @some_variable %>
7  <%= escape_once params[::some_cookie] %>
Static Analysis Detour
Static Analysis

Anything that can be determined about a program without actually executing it
“But Ruby is way too dynamic for that!”
eval(File.read(gets.strip))
We don’t have to know everything
Most of the Action Happens Here
Start Simple:
User Input in Views

<% params[:user][:name] %>
Next: From Controllers to Views

@user = params[:user]

<%= @user[:name] %>
Next: From Controllers to Views

Controller

```
@user = params[:user]
```

View

```
<%= @user[:name] %>
```
@user = params[:user]

<%=
  params[:user][:name]
%>
user = params[:user]
user_id = user[:id]
@current_user = User.find(user_id)
@current_user.update_attributes(user)
user = params[:user]
user_id = user[:id]
@current_user = User.find(user_id)
@current_user.update_attributes(user)
user = params[:user]
user_id = params[:id]
@current_user = User.find(user_id)
@current_user.update_attributes(user)
user = params[:user]
user_id = params[:id]
@current_user = User.find(user_id)
@current_user.update_attributes(user)
user = params[:user]
user_id = params[:id]
@current_user = User.find(params[:user][:id])
@current_user.update_attributes(user)
user = params[:user]
user_id = params[:id]
@current_user = User.find(params[:user][:id])
@current_user.update_attributes(user)
user = params[:user]
user_id = params[:id]
@current_user = User.find(params[:user][:id])
User.find(params[:user][:id]).update_attributes(user)
user = params[:user]
user_id = params[:id]
@current_user = User.find(params[:user][:id])
User.find(params[:user][:id]).update_attributes(user)
user = params[:user]
user_id = params[:id]
@current_user = User.find(params[:user][:id])
User.find(params[:user][:id]).update_attributes(params[:user])
user = params[:user]
user_id = params[:id]
@current_user = User.find(params[:user][:id])
User.find(params[:user][:id]).update_attributes(params[:user])
Brakeman Can Detect...

- Cross site scripting
- SQL injection
- Command injection
- Unrestricted mass assignment
- Unprotected redirects
- Unsafe file access
- Insufficient model validation

- Version-specific security issues
- Dangerous use of eval
- Dangerous use of send
- Default routes
- Dynamic render paths
- ...and more!
## Performance

<table>
<thead>
<tr>
<th>Application</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter Main App</td>
<td>&lt; 2m</td>
</tr>
<tr>
<td>nventory</td>
<td>~1m</td>
</tr>
<tr>
<td>(66c, 58m, 688t)</td>
<td></td>
</tr>
<tr>
<td>Redmine</td>
<td>~20s</td>
</tr>
<tr>
<td>(50c, 77m, 256t)</td>
<td></td>
</tr>
<tr>
<td>Typo</td>
<td>~5s</td>
</tr>
<tr>
<td>(34c, 47m, 113t)</td>
<td></td>
</tr>
</tbody>
</table>

Brakeman 1.6.0, Ruby 1.9.3-p125
Run Brakeman Anytime

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code
Run Brakeman Anytime

Write code  Run tests  Commit code  Push to CI server  Code review  QA Testing  Deploy code
Run Brakeman Anytime

Write code
Run tests
Commit code
Push to CI server
Code review
QA Testing
Deploy code
Run Brakeman Anytime

Write code
Run tests
Commit code
Push to CI server
Code review
QA Testing
Deploy code
Run Brakeman Anytime

Write code
Run tests
Commit code
Push to CI server
Code review
QA Testing
Deploy code
Brakeman + Jenkins
jenkins-ci.org
open source CI server
Brakeman + Jenkins
Brakeman + Jenkins

Warnings Trend

<table>
<thead>
<tr>
<th>All Warnings</th>
<th>New Warnings</th>
<th>Fixed Warnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>21</td>
<td>3</td>
</tr>
</tbody>
</table>

Summary

<table>
<thead>
<tr>
<th>Total</th>
<th>High Priority</th>
<th>Normal Priority</th>
<th>Low Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>41</td>
<td>12</td>
<td>3</td>
</tr>
</tbody>
</table>

Details

<table>
<thead>
<tr>
<th>Types</th>
<th>Total</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Restriction</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Command Injection</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Cross Site Request Forgery</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Cross Site Scripting</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Dangerous Eval</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Dangerous Send</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Default Routes</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Brakeman Programatically

require "brakeman"
Brakeman.run "myapp"
Run Brakeman Anytime

Write code  Run tests  Commit code  Push to CI server  Code review  QA Testing  Deploy code
Run Brakeman Anytime

Write code  Run tests  Commit code  Push to CI server  Code review  QA Testing  Deploy code
Run Brakeman Anytime

- Write code
- Run tests
- Commit code
- Push to CI server
- Code review
- QA Testing
- Deploy code
Brakeman + Rake

brakeman --rake
rake brakeman:run
Hardcore Mode

brakeman -z
Comparing Brakeman Results

brakeman -o report.json
brakeman --compare report.json
Brakeman...All the Time?

Write code
Run tests
Commit code
Push to CI server
Code review
QA Testing
Deploy code
Brakeman...All the Time?

Write code | Run tests | Commit code | Push to CI server | Code review | QA Testing | Deploy code

Save code
Fast Rescanning

Brakeman supports fast rescanning of changed files*
Fast Rescanning

*If scan is kept in memory*
Brakeman + Guard

group :development do
gem 'guard-brakeman'
end
Brakeman + Guard

guard init brakeman
guard
Brakeman + Guard Demo

http://www.youtube.com/watch?v=CMgYcr9_ONs
Caveats
warnings != vulnerabilities
zero warnings
does not mean
zero vulnerabilities
Brakeman is not omniscient
Supports

- Rails 2.x & 3.x
- Ruby 1.8.7 & 1.9.x
- JRuby