

# Automated Detection of Security Flaws in Ruby on Rails Applications

Justin Collins  
OWASP LA  
May 25<sup>th</sup>, 2011



# Goal

Automated, hands-free vulnerability reporting  
for Ruby on Rails web applications



# Ruby on Rails Web Framework

- Uses the Ruby programming language
- Model-View-Controller
  - *Model*: data storage and access
  - *View*: data presentation
  - *Controller*: business logic
- Convention over configuration
  - Many assumptions that lead to default behavior



<http://rubyonrails.org/>



# Simple Model

```
class User < ActiveRecord::Base  
end
```



# Simple Controller

```
class HomeController < ApplicationController  
  def index  
  end  
end
```



# Simple Controller

Defaults to `mysite.com/home`

```
class HomeController < ApplicationController
  def index
  end
end
```



# Simple Controller

Defaults to `mysite.com/home`

```
class HomeController < ApplicationController
  def index
  end
end
```

Will render view from `app/views/home/index.*`



# Simple View Using ERB

```
<h2>Home</h2>
```

```
Welcome, <%= cookie[:user_name] %>!
```





# Tools

## Brakeman

Static analysis vulnerability scanner for Ruby on Rails applications

`github.com/presidentbeef/brakeman`

## Jenkins/Hudson

Continuous integration and job monitoring system

`jenkins-ci.org`



# Why Static Analysis?

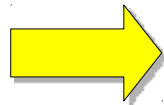
- No deployment necessary
  - No configuration, no database, no web server
- Usable at any stage of development
  - And by developers
- No reliance on link crawling
- Inside view of application



# What Brakeman Does

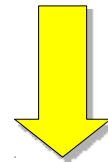
Parse Rails App Code

```
<h2><%= l(:label_home)
%></h2><div
class="splitcontentleft"
> <%= textilizable
Setting.welcome_text %>
<% if @news.any? %>
<div class="news box">
  <h3><
%=l(:label_news_latest)
%></h3> <%=
render :partial =>
'news/news', :collection
=> @news %>
```

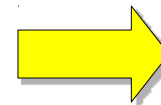


S-Expressions

```
s(:call, nil, puts,
(:arglist,
s(:call, nil, 'hello',
s(:arglist))))
```



Cleanup and Simplify



Generate Report



Run Checks



# Variable Propagation

```
class HomeController < ApplicationController
  def index
    user_id = params[:id]
    user = User.find(user_id)
    @name = user.name
  end
end
```

---

```
<h2>Home</h2>
```

```
Welcome, <%= @name %>!
```

# Variable Propagation

```
class HomeController < ApplicationController
  def index
    user_id = params[:id]
    user = User.find(params[:id])
    @name = user.name
  end
end
```

---

```
<h2>Home</h2>
```

```
Welcome, <%= @name %>!
```

# Variable Propagation

```
class HomeController < ApplicationController
  def index
    user_id = params[:id]
    user = User.find(params[:id])
    @name = User.find(params[:id]).name
  end
end
```

---

```
<h2>Home</h2>
```

```
Welcome, <%= @name %>!
```

# Variable Propagation

```
class HomeController < ApplicationController
  def index
    user_id = params[:id]
    user = User.find(params[:id])
    @name = User.find(params[:id]).name
  end
end
```

---

<h2>Home</h2>

Welcome, <%= User.find(params[:id]).name %>!

# Brakeman Vulnerability Detection

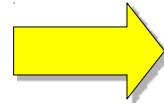
- SQL Injection
- Command Injection
- Cross site scripting
- Unprotected redirects
- Unsafe file access
- Default routes
- Insufficient validation on model input
- Unsafe configurations
- ...and more



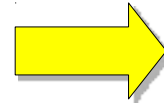


# What Jenkins Does

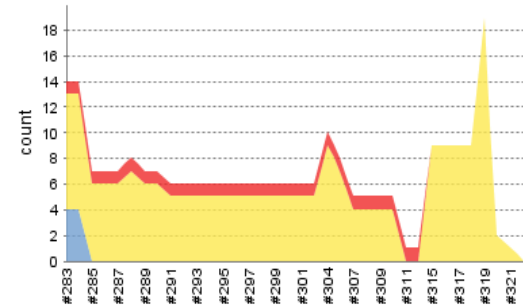
Monitor Conditions



Run Job



Aggregate Results

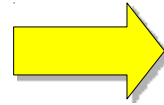


# What Jenkins Does

Monitor Conditions



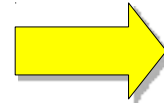
```
git push  
svn commit
```



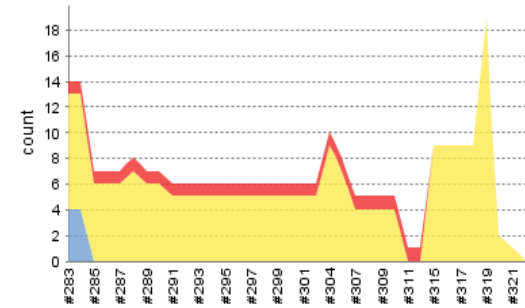
Run Job



Brakeman



Aggregate Results



Security Warnings



# More Information

- Ruby  
<http://ruby-lang.org>
- Ruby on Rails  
<http://rubyonrails.org>
- Ruby on Rails Security Guide  
<http://guides.rubyonrails.org/security.html>
- Brakeman  
<http://github.com/presidentbeef/brakeman>
- Jenkins  
<http://jenkins-ci.org>
- Brakeman plugin for Jenkins  
<http://github.com/presidentbeef/brakeman-jenkins-plugin>

