GOTHAM
DIGITAL • SCIENCE

Daniel Peláez
dpelaez@gdssecurity.com

Security Goodness with Ruby On Rails

SOURCE BARCELONA
16th November 2011

AGENDA

- Who Am I?
- Brief Introduction to Rails
- How Secure is Ruby On Rails?
- Auditing Applications
- Building Secure Rails WebSites

Best practices, tools, security APIs.
How to identify and fix common vulnerabilities.

WHO AM I?

IT Security Consultant at Gotham Digital Science (GDS)

- Another crazy Spaniard who recently moved to **London**
- I have some experience with Rails & also with Security:
  - Pentests
  - Source Code Reviews
  - Consulting
  - Blablabla :)

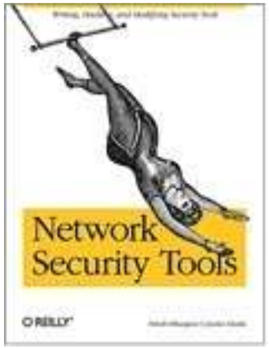
About GDS

- **Gotham Digital Science** (GDS) is an international security services company specializing in Application and Network Infrastructure security, and Information Security Risk Management. GDS clients number among the largest financial services institutions and software development companies in the world.

- Offices in **London** and **New York City**

About GDS

- **Tools & Papers:**
  - ***Padbuster***, Blazentoo, GwtEnum … etc
- **Publications with GDS Contributing Authors:**

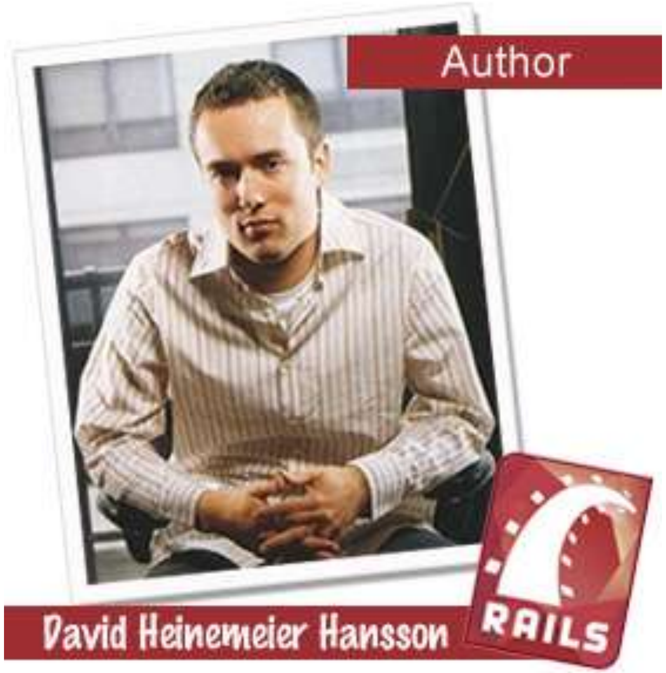


Overview of what is Rails

BRIEF INTRODUCTION

SECURITY GOODNESS WITH RUBY ON RAILS

Brief Introduction to Rails



RAILS™ Ruby on Rails

37signals

Web development that doesn't hurt

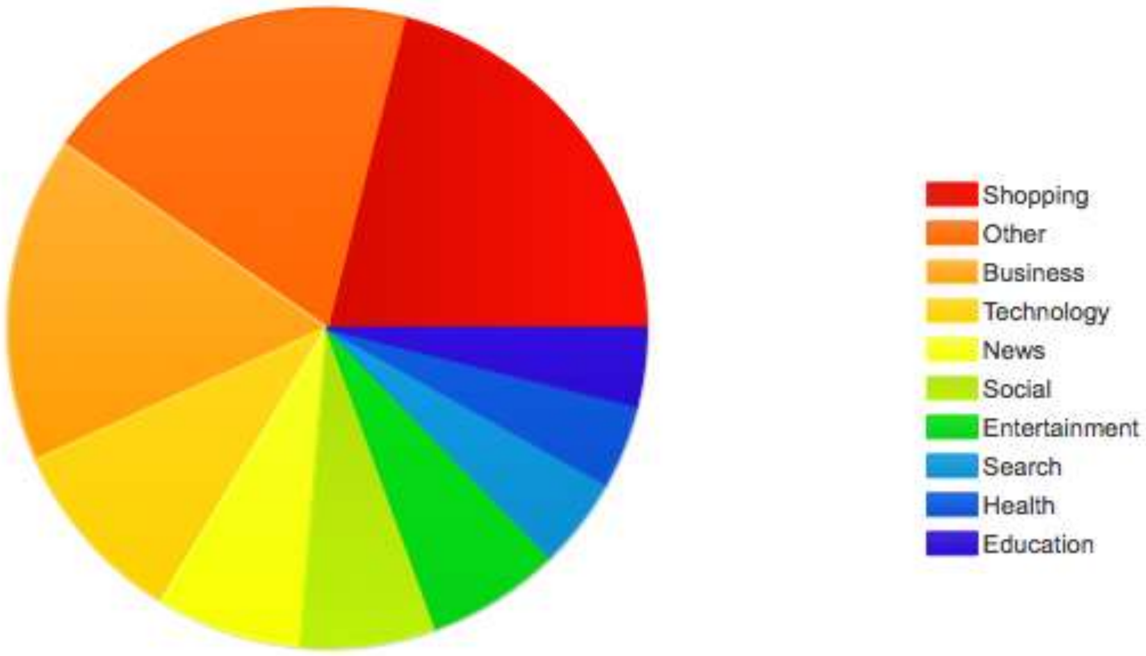
Ruby on Rails® is an open-source web framework that's optimized for programmer happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration.

www.rubyonrails.org

Brief Introduction to Rails

- WebSite Industries

Ruby on Rails Top Ten Website Industry Distribution in the Top 100,000 Sites



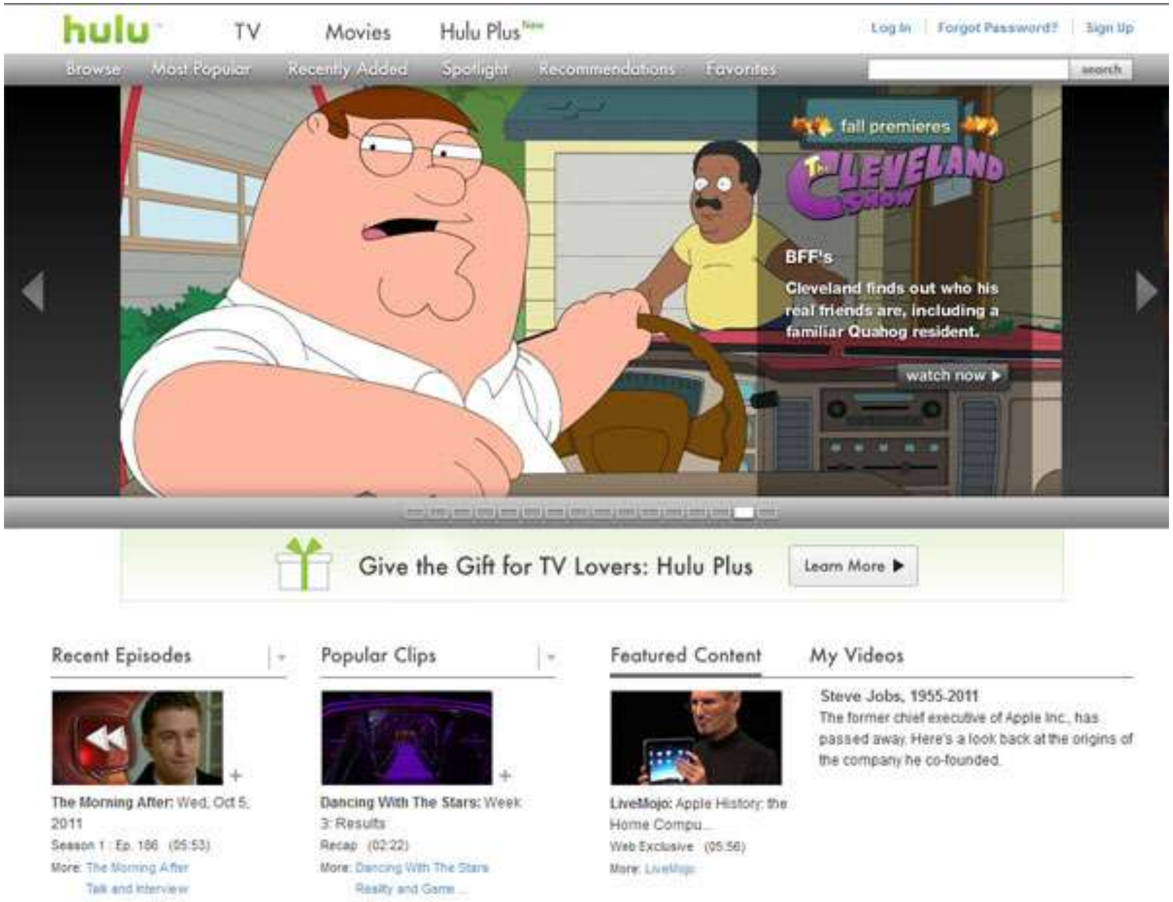
BRIEF INTRODUCTION TO RAILS

- Who uses Rails?

  - Twitter (In the early days)
  - Groupon
  - **Linkedin**
  - **Github**
  - **Basecamp**
  - SlideShare
  - Funny or Die
  - Scribd
  - CrunchBase
  - **Hulu**
  - Zendesk
  - **YellowPages**
  - OneHub
  - Jobster
  - Heroku
  - Rackspace
  - Engine Yard
  - Shopify

Brief Introduction to Rails

- Hulu.com



BFF's

Cleveland finds out who his real friends are, including a familiar Quahog resident.

Give the Gift for TV Lovers: Hulu Plus

The Morning After: Wed, Oct 5, 2011
Season 1 : Ep. 186 (05:53)
More: The Morning After
Talk and Interview

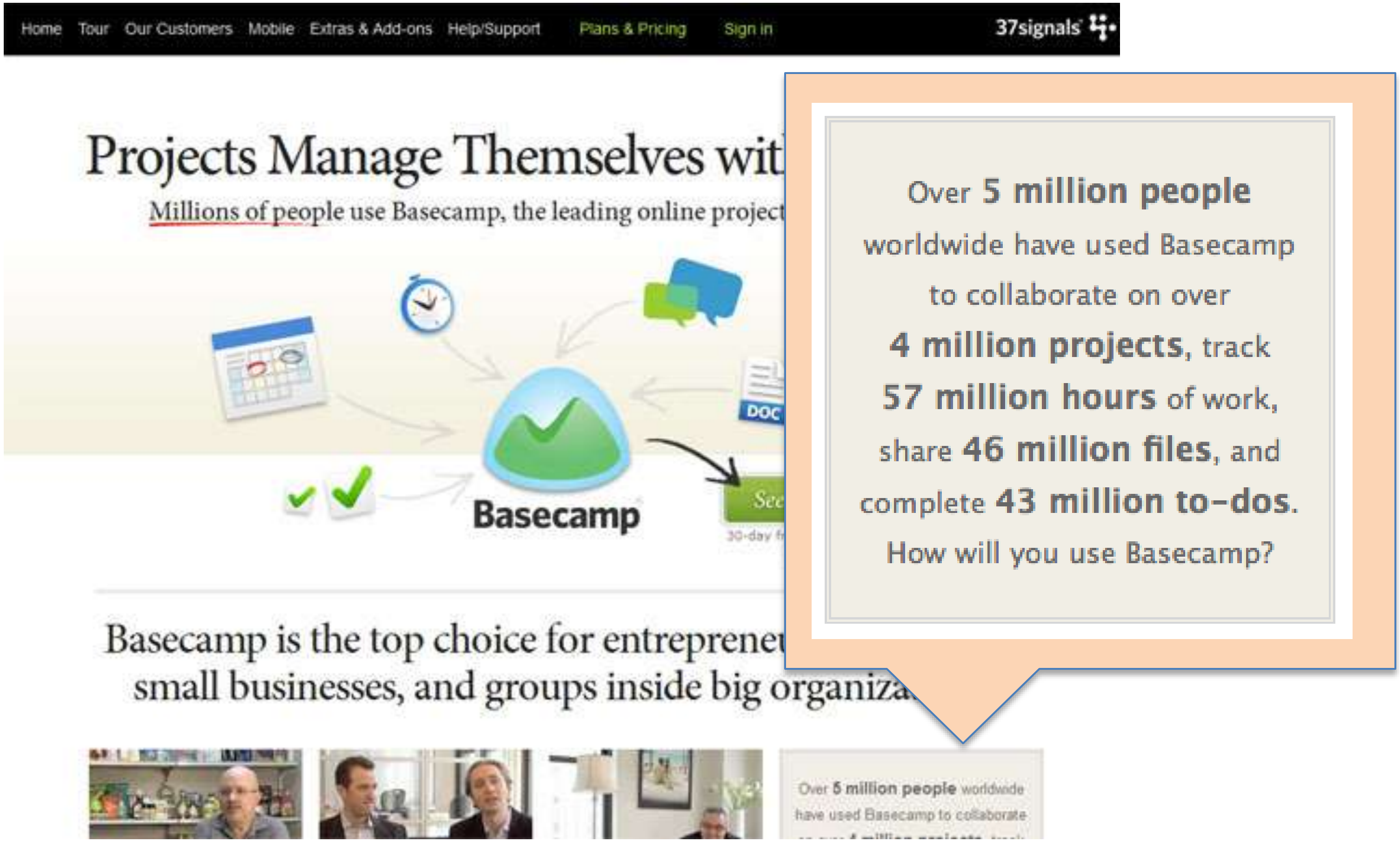
Dancing With The Stars: Week 3: Results
Recap (02:22)
More: Dancing With The Stars
Reality and Game ...

LiveMojo: Apple History: the Home Compu...
Web Exclusive (05:56)
More: LiveMojo

Steve Jobs, 1955-2011
The former chief executive of Apple Inc., has passed away. Here's a look back at the origins of the company he co-founded.

Brief Introduction to Rails

- basecamphq.com



Projects Manage Themselves wit

Millions of people use Basecamp, the leading online project

Basecamp is the top choice for entreprene

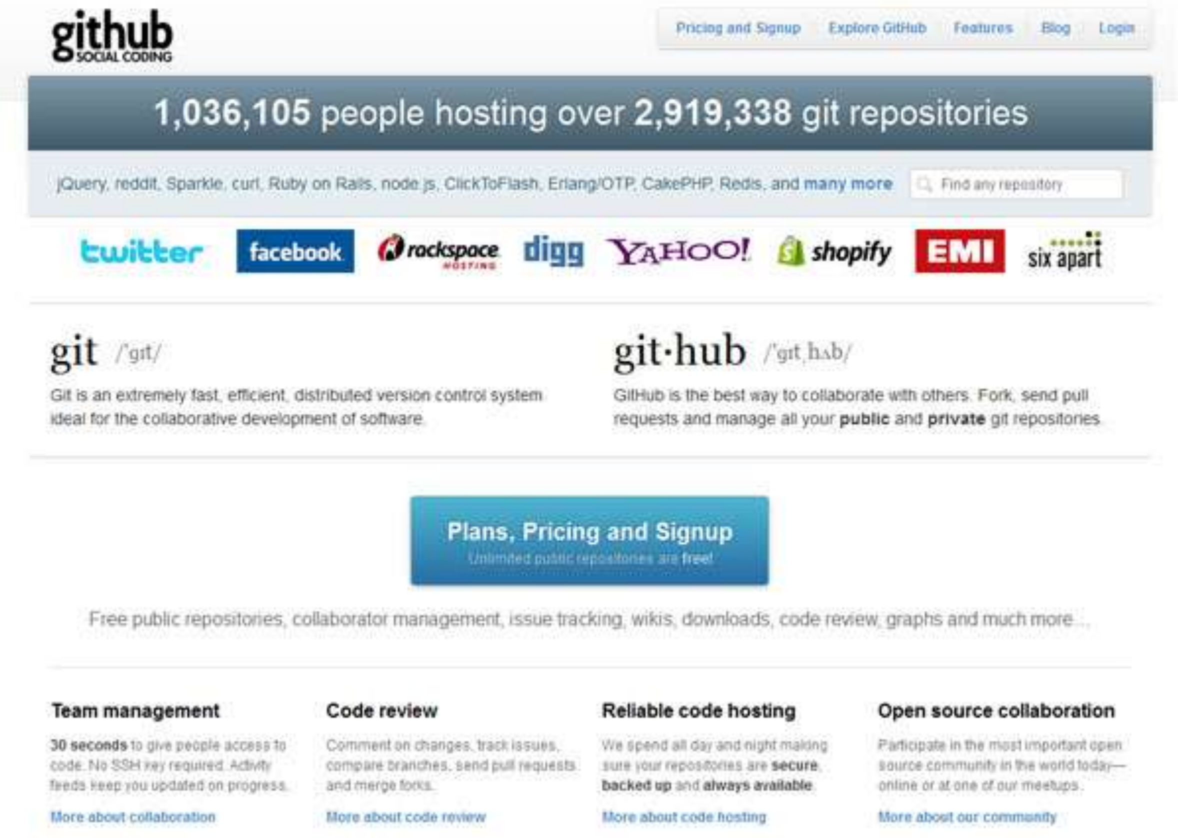
small businesses, and groups inside big organiza

Over **5 million people** worldwide have used Basecamp to collaborate on over **4 million projects**, track **57 million hours** of work, share **46 million files**, and complete **43 million to-dos**. How will you use Basecamp?

Over 5 million people worldwide have used Basecamp to collaborate

Brief Introduction to Rails

- GitHub.com



1,036,105 people hosting over **2,919,338** git repositories

jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, and **many more**

git /ˈgɪt/

Git is an extremely fast, efficient, distributed version control system ideal for the collaborative development of software.

git·hub /ˈgɪtˌhʌb/

GitHub is the best way to collaborate with others. Fork, send pull requests and manage all your **public** and **private** git repositories.

Unlimited public repositories are free!

Free public repositories, collaborator management, issue tracking, wikis, downloads, code review, graphs and much more...

Team management

30 seconds to give people access to code. No SSH key required. Activity feeds keep you updated on progress.

More about collaboration

Code review

Comment on changes, track issues, compare branches, send pull requests and merge forks.

More about code review

Reliable code hosting

We spend all day and night making sure your repositories are **secure**, **backed up** and **always available**.

More about code hosting

Open source collaboration

Participate in the most important open source community in the world today—online or at one of our meetups.

More about our community

Philosophy and Design

BRIEF INTRODUCTION

SECURITY GOODNESS WITH RUBY ON RAILS

Brief Introduction to Rails

- Ruby
  - Rails
  - Sinatra
  - Merb*
- PHP
  - Zend
  - CakePHP
  - Symfony
  - Zoop
  - Akelos
- Java
  - Struts
  - Spring
  - Stripes
  - Hivemind
  - JBoss
- Python
  - Django
  - Pylons
  - Zope
  - TurboGears

FRAMEWORK

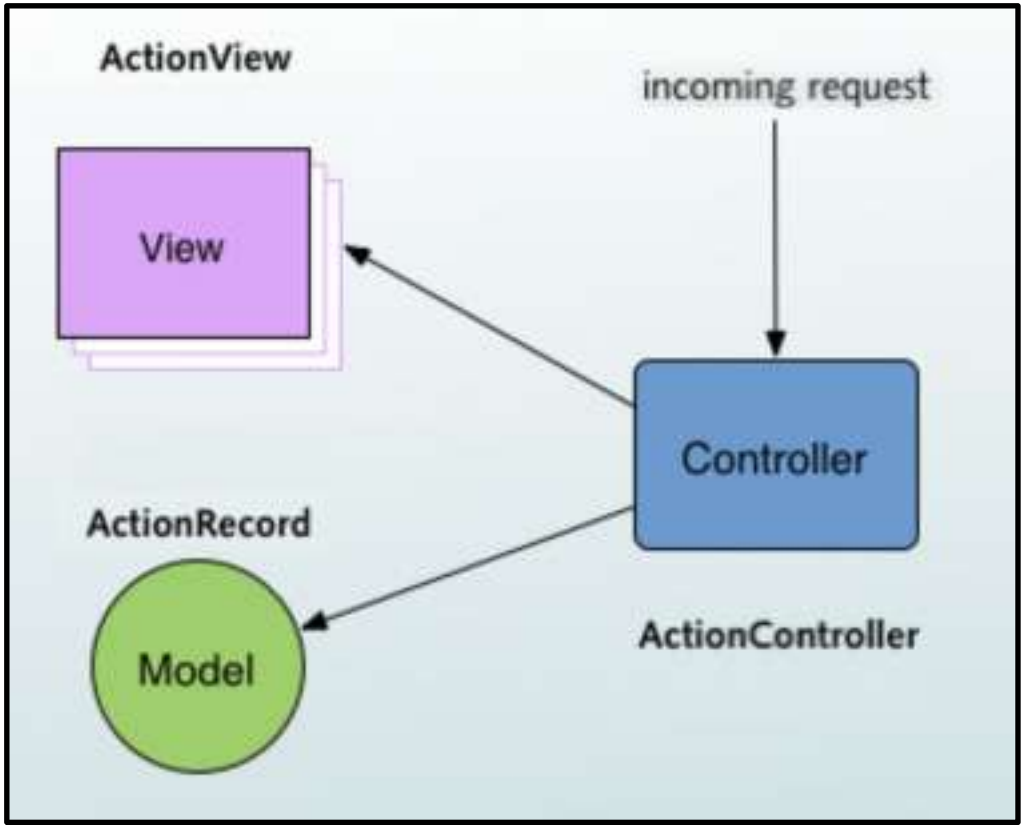
Model-View-Controller (MVC) architecture pattern

CONVENTION OVER CONFIGURATION (COC)

DON’T REPEAT YOURSELF (DRY)

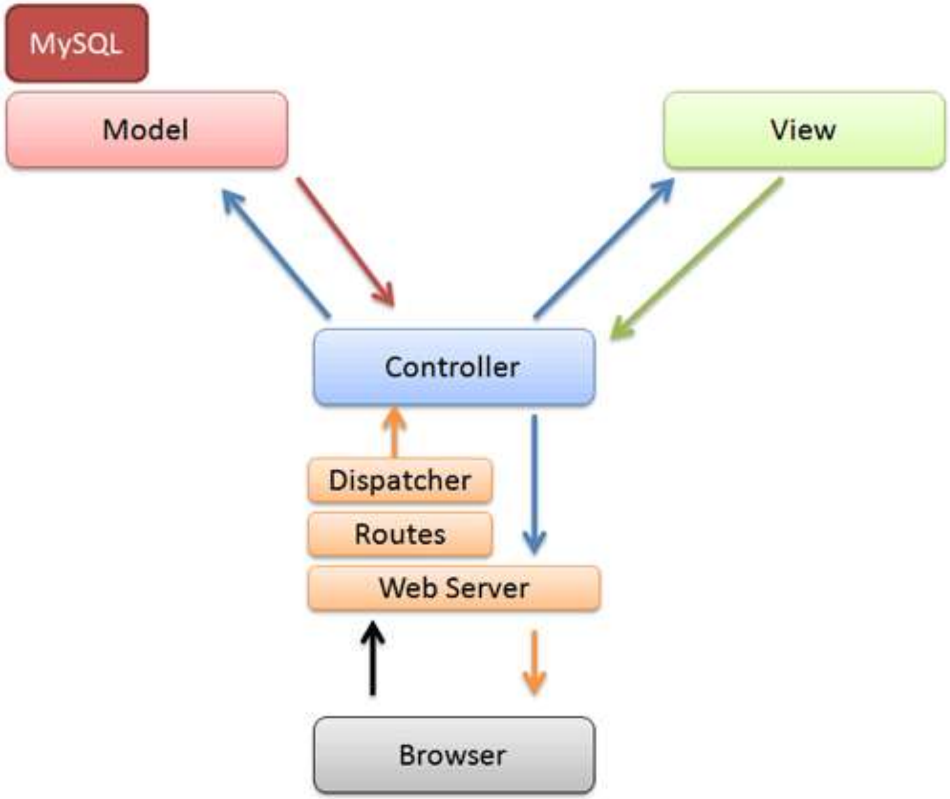
BRIEF INTRODUCTION TO RAILS

Rails Components & MVC



BRIEF INTRODUCTION TO RAILS

Rails Components & MVC



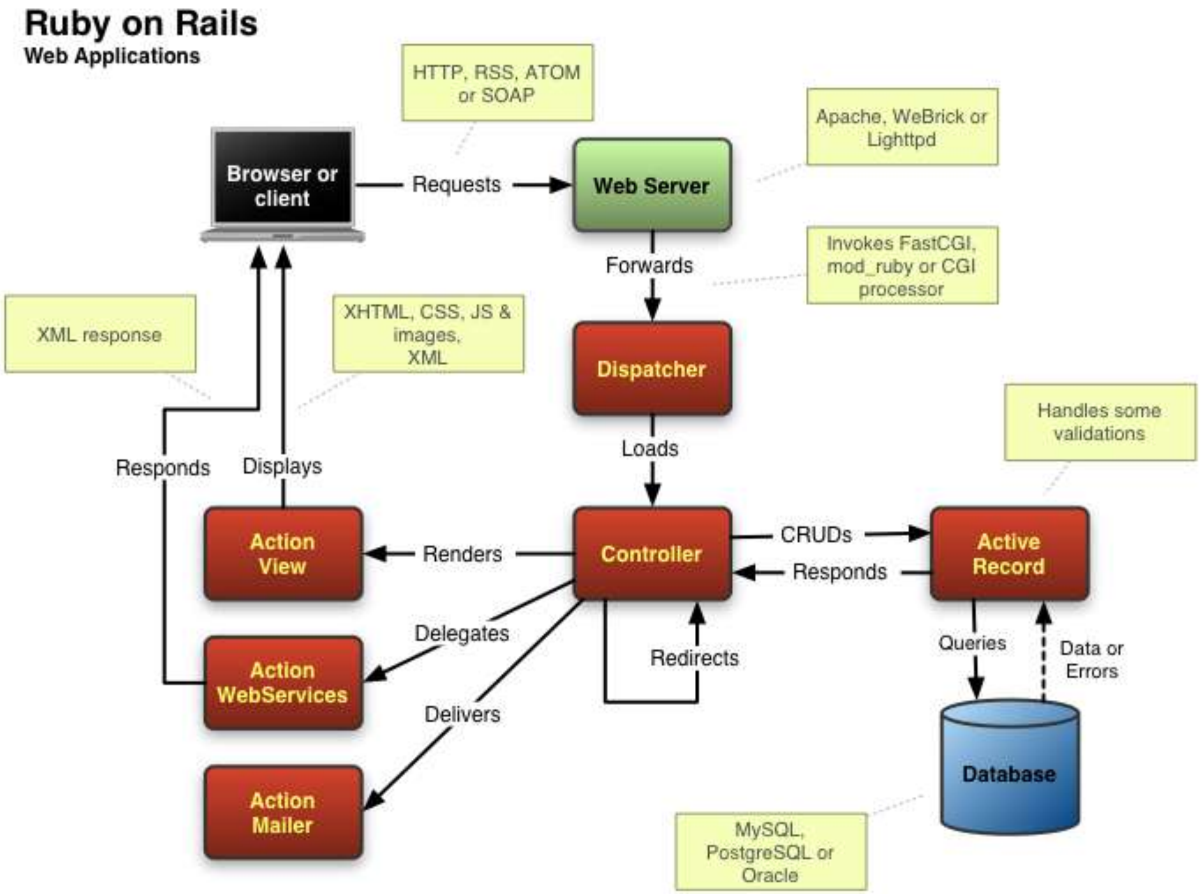
BRIEF INTRODUCTION TO RAILS

Model-View-Controller (MVC) architecture pattern

- **Action Controller**
  - Processes incoming requests to a Rails application, extracts parameters, and dispatches them to the intended action.
  - Services provided by Action Controller include session management, template rendering, and redirect management.
- **Action View**
  - It can create both HTML and XML output by default.
  - Manages rendering templates, including nested and partial templates, and includes built-in AJAX support.
- **Action Dispatch**
  - Handles **routing** of web requests and dispatches them as you want, either to your application or any other Rack application.
- **Active Record**
  - It provides **database independence**, basic CRUD functionality, advanced finding capabilities, and the ability to relate models to one another, among other services.
- **Active Model**
  - Interface between the Action Pack gem services and Object Relationship Mapping gems such as Active Record. Active Model allows Rails to utilize other ORM frameworks in place of Active Record.

BRIEF INTRODUCTION TO RAILS

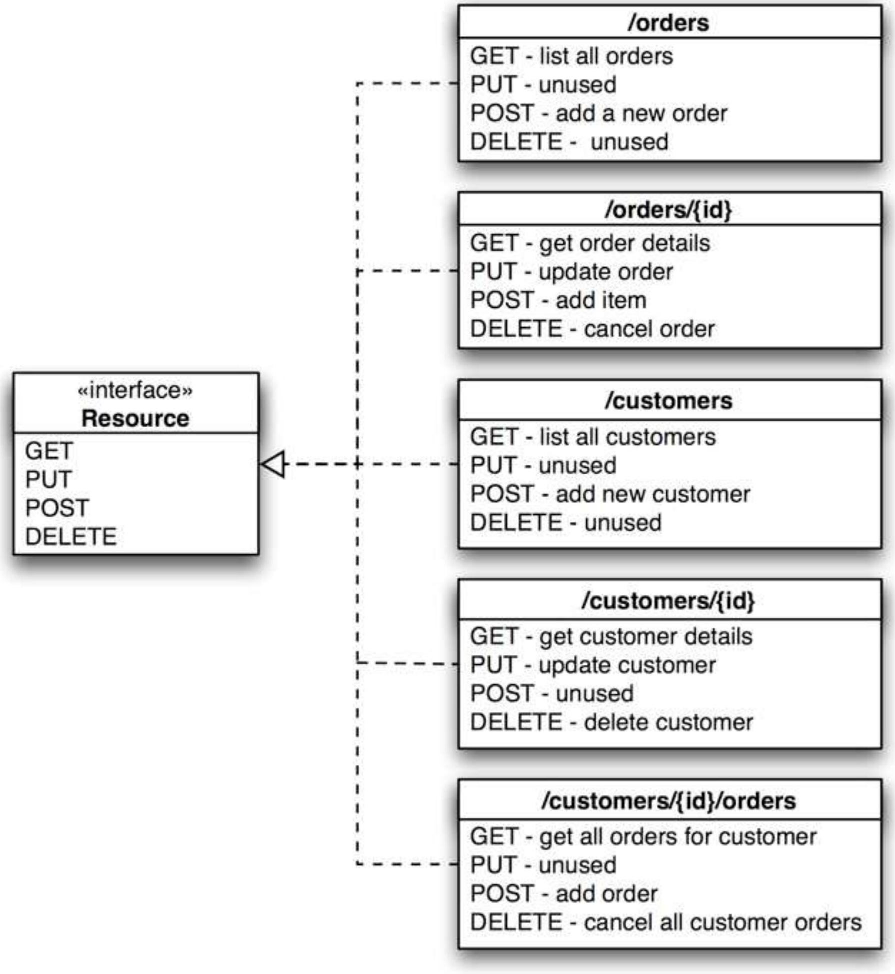
Generic Rails Architecture Diagram



Brief Introduction to Rails

- **REST (Representational State Transfer)**
  - Using resource identifiers such as URLs to represent resources.
  - Transferring representations of the state of that resource between system components.
  - GET /orders/17
  - PUT /orders/26
  - POST /orders/17
  - DELETE /orders/26

Brief Introduction to Rails



/orders
GET - list all orders
PUT - unused
POST - add a new order
DELETE - unused

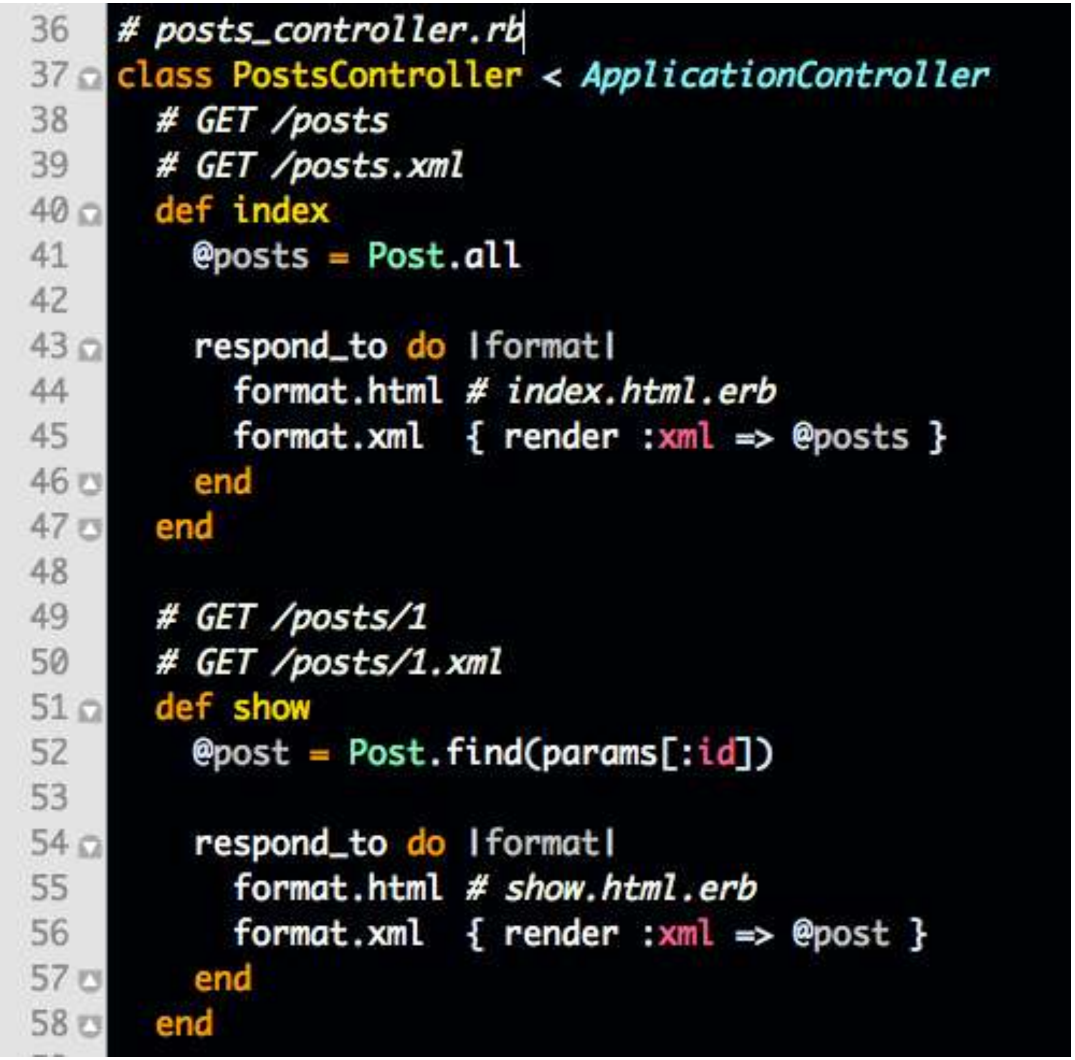
/orders/{id}
GET - get order details
PUT - update order
POST - add item
DELETE - cancel order

/customers
GET - list all customers
PUT - unused
POST - add new customer
DELETE - unused

/customers/{id}
GET - get customer details
PUT - update customer
POST - unused
DELETE - delete customer

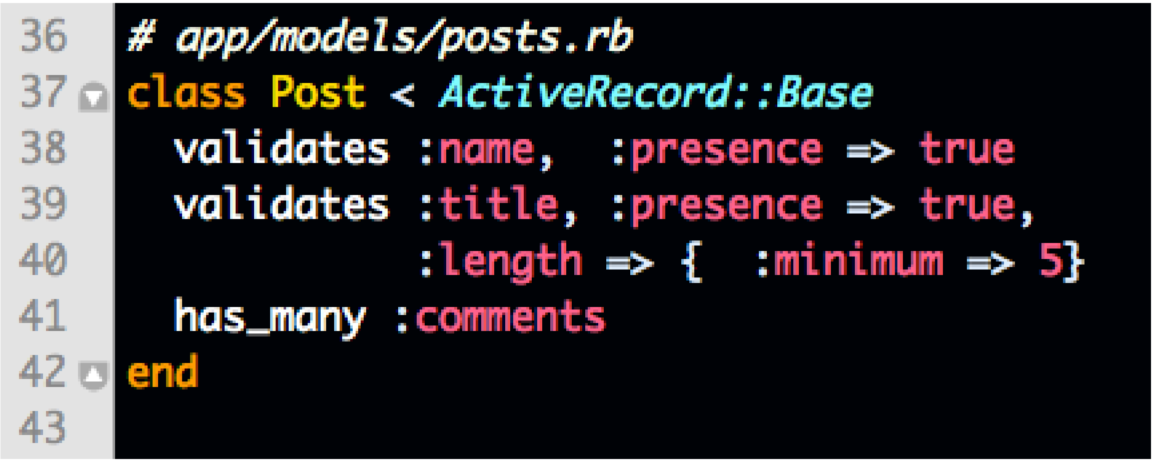
/customers/{id}/orders
GET - get all orders for customer
PUT - unused
POST - add order
DELETE - cancel all customer orders

BRIEF INTRODUCTION TO RAILS



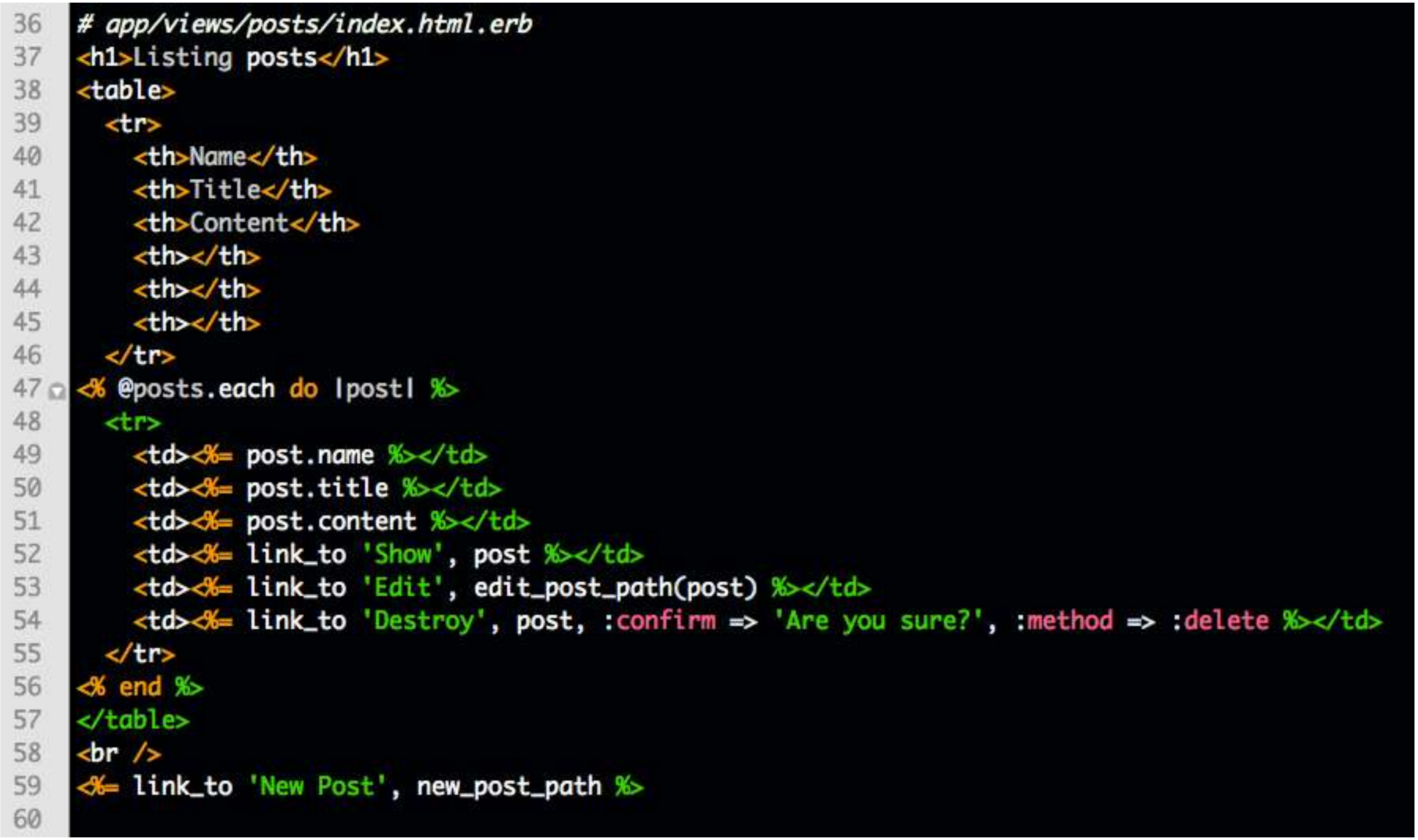
```ruby
# posts_controller.rb
class PostsController < ApplicationController
  # GET /posts
  # GET /posts.xml
  def index
    @posts = Post.all

    respond_to do |format|
      format.html # index.html.erb
      format.xml  { render :xml => @posts }
    end
  end

  # GET /posts/1
  # GET /posts/1.xml
  def show
    @post = Post.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.xml  { render :xml => @post }
    end
  end
```

Brief Introduction to Rails



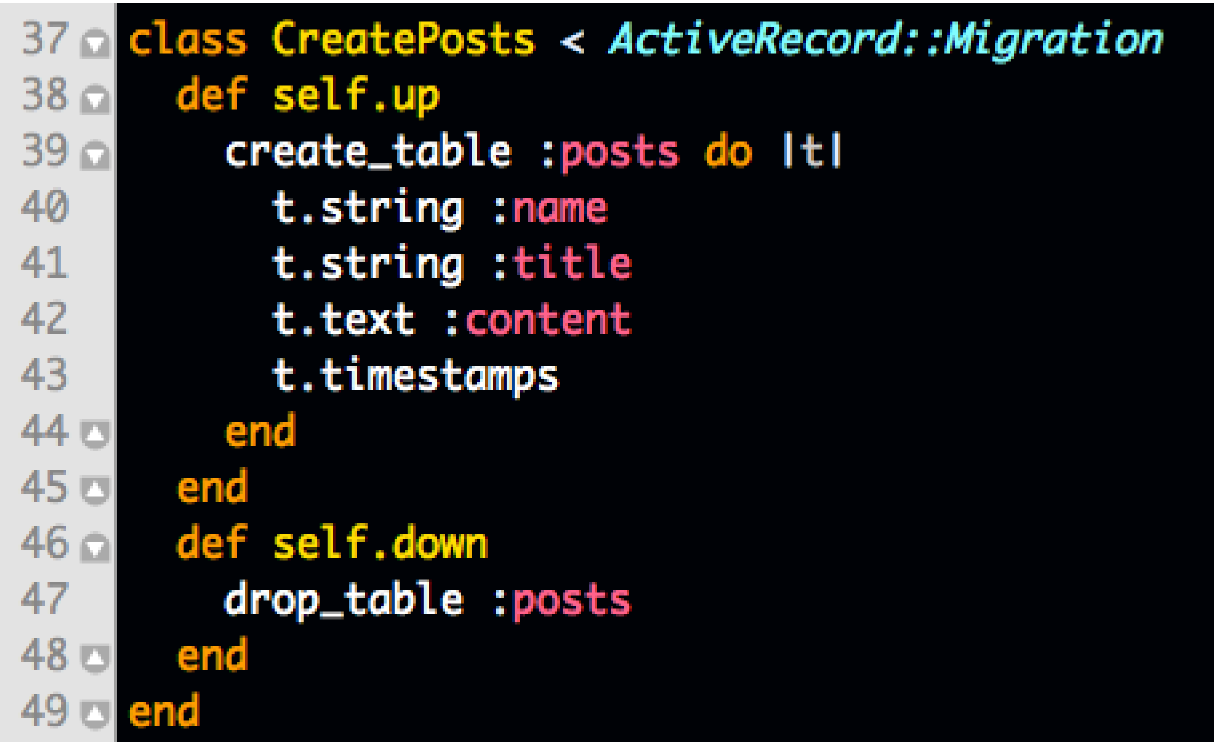
```
# app/models/posts.rb
class Post < ActiveRecord::Base
  validates :name,  :presence => true
  validates :title, :presence => true,
            :length => {  :minimum => 5}
  has_many :comments
end
```

BRIEF INTRODUCTION TO RAILS



```erb
# app/views/posts/index.html.erb
<h1>Listing posts</h1>
<table>
  <tr>
    <th>Name</th>
    <th>Title</th>
    <th>Content</th>
    <th></th>
    <th></th>
    <th></th>
  </tr>
<% @posts.each do |post| %>
  <tr>
    <td><%= post.name %></td>
    <td><%= post.title %></td>
    <td><%= post.content %></td>
    <td><%= link_to 'Show', post %></td>
    <td><%= link_to 'Edit', edit_post_path(post) %></td>
    <td><%= link_to 'Destroy', post, :confirm => 'Are you sure?', :method => :delete %></td>
  </tr>
<% end %>
</table>
<br />
<%= link_to 'New Post', new_post_path %>
```

Brief Introduction to Rails



```ruby
class CreatePosts < ActiveRecord::Migration
  def self.up
    create_table :posts do |t|
      t.string :name
      t.string :title
      t.text :content
      t.timestamps
    end
  end
  def self.down
    drop_table :posts
  end
end
```

Tools – Vulnerabilities - Recommendations

AUDITING APPLICATIONS

SECURITY GOODNESS WITH RUBY ON RAILS

AUDITING RAILS APPLICATIONS

The Basic Defense Points

- Authentication:
  - Is the application enforcing an acceptable password policy for users?
  - Can the authentication process be bypassed?
- Authorization:
  - Does the application have authorization checks for all default and custom actions?
- Data Protection:
  - Are sensitive database fields encrypted or hashed?
  - Is TLS / SSL enforced during the transmission of sensitive information such as passwords or credit card numbers?
- Input Validation & Sanitization:
  - Is all input validated on the server?
  - When displaying information, are we sanitizing the output?

AUDITING RAILS APPLICATIONS

Information Leaks: How to Identify Rails WebSites

- **MONGREL**

  Server: Mongrel 1.1.5

- **APACHE**

  Server: Apache/1.3.34 (Unix) mod_deflate/1.0.21 mod_fastcgi/2.4.2 mod_ssl/2.8.25 OpenSSL/0.9.7e-p1

- **NGINX**

  X-Powered-By: Phusion Passenger (mod_rails/mod_rack) 3.0.7

  X-Runtime: 0.008653

  Server: nginx/1.0.0 + **Phusion Passenger** 3.0.7 (mod_rails/mod_rack)

Removing HTTP Headers

- **APACHE**

  Add these lines to httpd.conf

  - Header always unset "X-Powered-By"
  - Header always unset "X-Runtime"
  - Header always unset "Server"

- **NGINX**

  Add this directive to HttpHeadersMoreModule

  - more_clear_headers Server X-Powered-By X-Runtime;

AUDITING RAILS APPLICATIONS

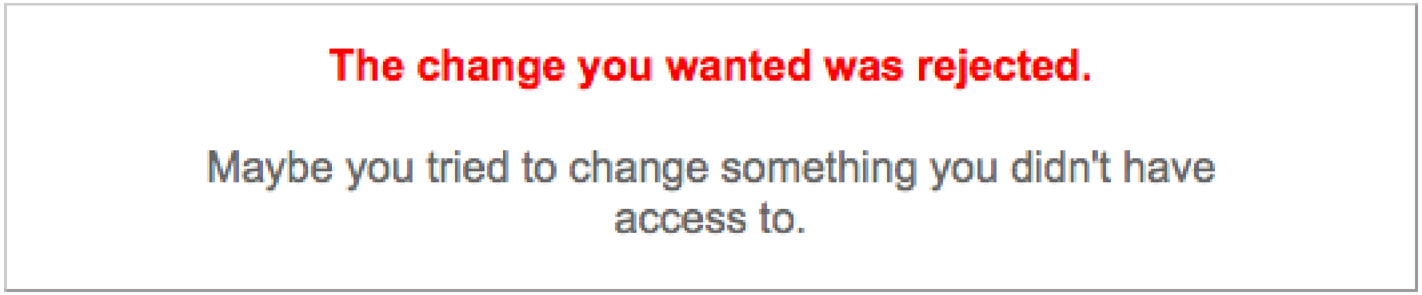
Information Leaks: How to Identify Rails WebSites

- Default Static Files:
  - /javascripts/application.js
  - /javascripts/prototype.js
  - /stylesheets/application.css
  - /images/rails.png
- Pretty URLs (RESTful):
  - /posts/32/edit
  - /project/create
  - /folders/delete/54
  - /users/81

AUDITING RAILS APPLICATIONS

Information Leaks: How to Identify Rails WebSites

- Different default pages depending on Rails version
- Default templates for 404 and 500 status pages
- 422.html only in applications generated with Rails >= 2.0



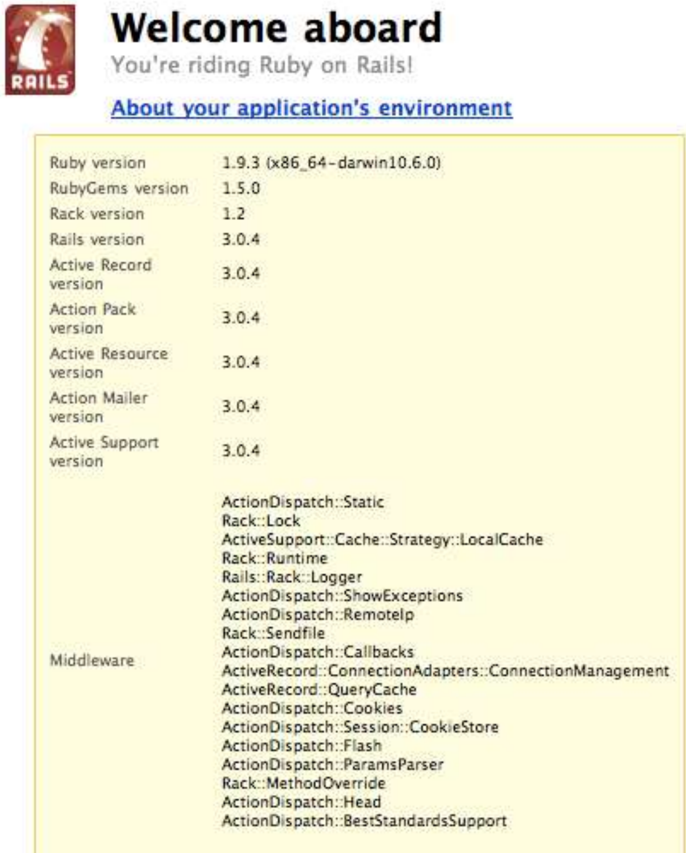
The change you wanted was rejected.

Maybe you tried to change something you didn't have access to.

Auditing Rails Applications

Information Leaks: How to Identify Rails WebSites

- Stack Traces / error pages

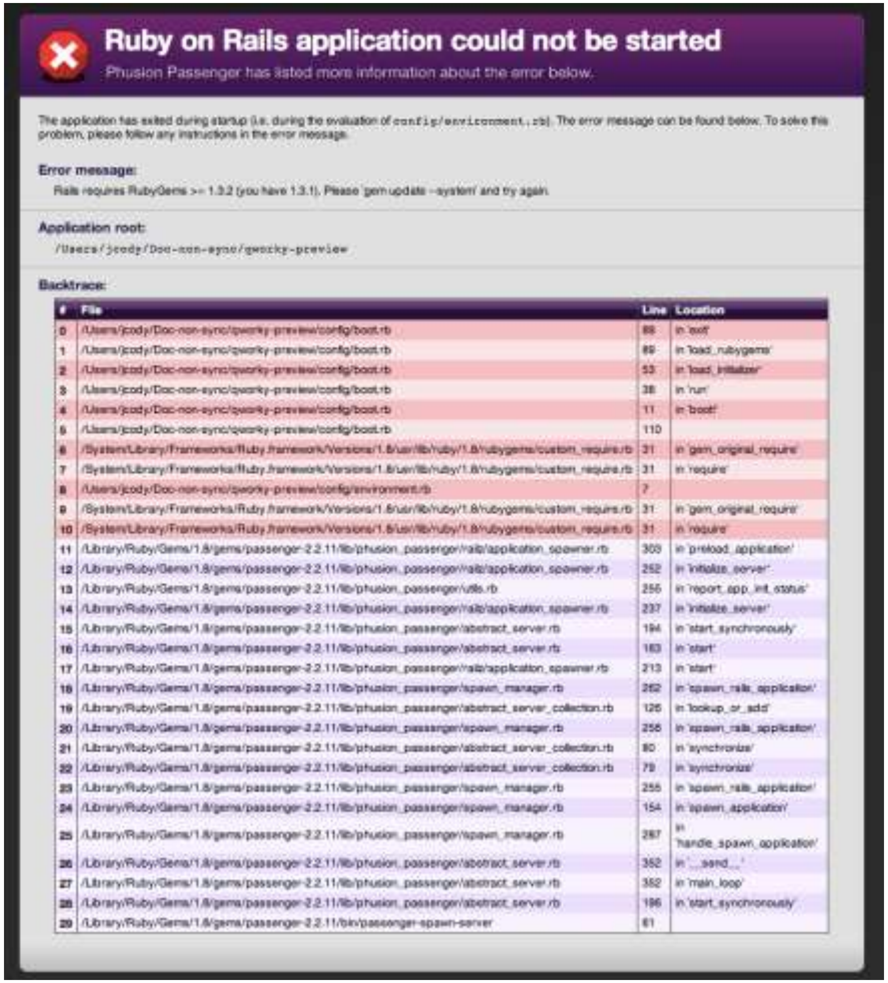


Welcome aboard

You're riding Ruby on Rails!

About your application's environment

Ruby version	1.9.3 (x86_64-darwin10.6.0)
RubyGems version	1.5.0
Rack version	1.2
Rails version	3.0.4
Active Record version	3.0.4
Action Pack version	3.0.4
Active Resource version	3.0.4
Action Mailer version	3.0.4
Active Support version	3.0.4
Middleware	ActionDispatch::Static Rack::Lock ActiveSupport::Cache::Strategy::LocalCache Rack::Runtime Rails::Rack::Logger ActionDispatch::ShowExceptions ActionDispatch::RemoteIp Rack::Sendfile ActionDispatch::Callbacks ActiveRecord::ConnectionAdapters::ConnectionManagement ActiveRecord::QueryCache ActionDispatch::Cookies ActionDispatch::Session::CookieStore ActionDispatch::Flash ActionDispatch::ParamsParser Rack::MethodOverride ActionDispatch::Head ActionDispatch::BestStandardsSupport



Ruby on Rails application could not be started

Phusion Passenger has listed more information about the error below.

The application has exited during startup (i.e. during the evaluation of `config/environment.rb`). The error message can be found below. To solve this problem, please follow any instructions in the error message.

Error message:

Rails requires RubyGems >= 1.3.2 (you have 1.3.1). Please `gem update --system` and try again.

Application root:

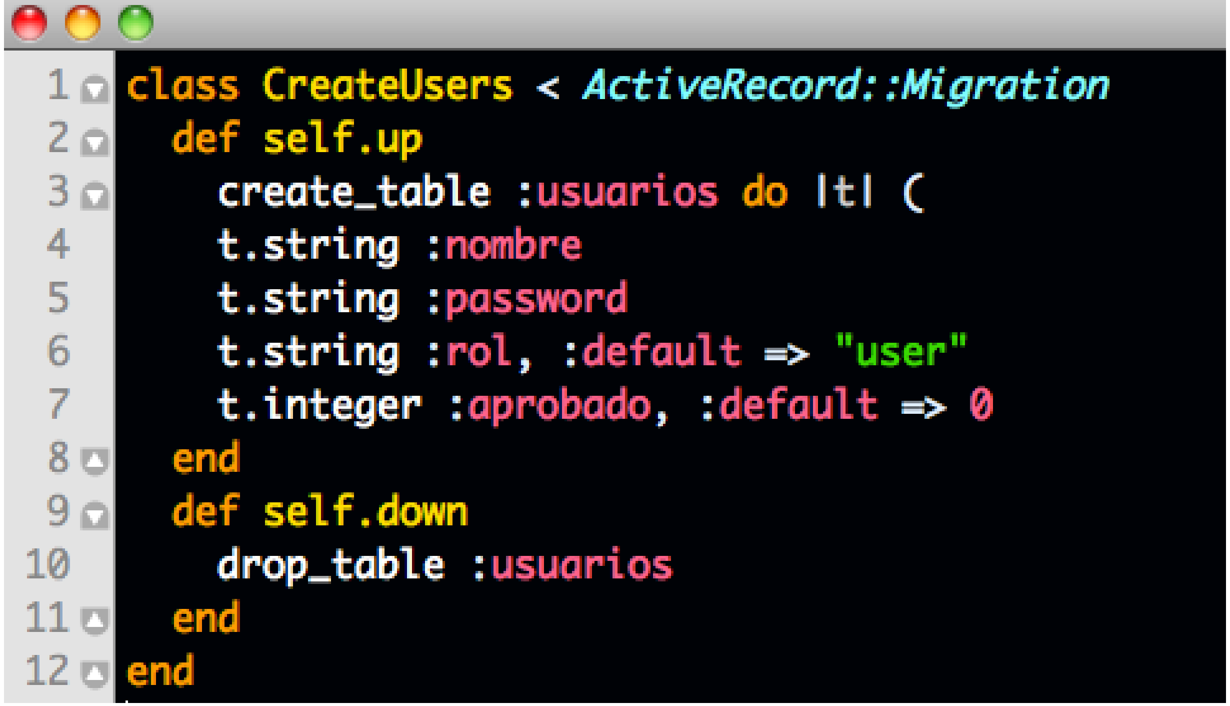
`/Users/jcody/Doc-non-sync/qworky-preview`

Backtrace:

#	File	Line	Location
0	/Users/jcody/Doc-non-sync/qworky-preview/config/boot.rb	88	in 'exit'
1	/Users/jcody/Doc-non-sync/qworky-preview/config/boot.rb	89	in 'load_rubygems'
2	/Users/jcody/Doc-non-sync/qworky-preview/config/boot.rb	53	in 'load_initializer'
3	/Users/jcody/Doc-non-sync/qworky-preview/config/boot.rb	38	in 'run'
4	/Users/jcody/Doc-non-sync/qworky-preview/config/boot.rb	11	in 'boot!'
5	/Users/jcody/Doc-non-sync/qworky-preview/config/boot.rb	110	
6	/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/lib/ruby/1.8/rubygems/custom_require.rb	31	in 'gem_original_require'
7	/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/lib/ruby/1.8/rubygems/custom_require.rb	31	in 'require'
8	/Users/jcody/Doc-non-sync/qworky-preview/config/environment.rb	7	
9	/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/lib/ruby/1.8/rubygems/custom_require.rb	31	in 'gem_original_require'
10	/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/lib/ruby/1.8/rubygems/custom_require.rb	31	in 'require'
11	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/railz/application_spawner.rb	303	in 'preload_application'
12	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/railz/application_spawner.rb	252	in 'initialize_server'
13	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/utils.rb	256	in 'report_app_init_status'
14	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/railz/application_spawner.rb	237	in 'initialize_server'
15	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/abstract_server.rb	194	in 'start_synchronously'
16	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/abstract_server.rb	183	in 'start'
17	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/railz/application_spawner.rb	213	in 'start'
18	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/spawn_manager.rb	262	in 'spawn_rails_application'
19	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/abstract_server_collection.rb	126	in 'lookup_or_add'
20	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/spawn_manager.rb	258	in 'spawn_rails_application'
21	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/abstract_server_collection.rb	80	in 'synchronize'
22	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/abstract_server_collection.rb	79	in 'synchronize'
23	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/spawn_manager.rb	255	in 'spawn_rails_application'
24	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/spawn_manager.rb	154	in 'spawn_application'
25	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/spawn_manager.rb	287	in 'handle_spawn_application'
26	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/abstract_server.rb	352	in '__send__'
27	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/abstract_server.rb	352	in 'main_loop'
28	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/lib/phusion_passenger/abstract_server.rb	196	in 'start_synchronously'
29	/Library/Ruby/Gems/1.8/gems/passenger-2.2.11/bin/passenger-spawn-server	61	

Vulnerabilities: Mass Assignment

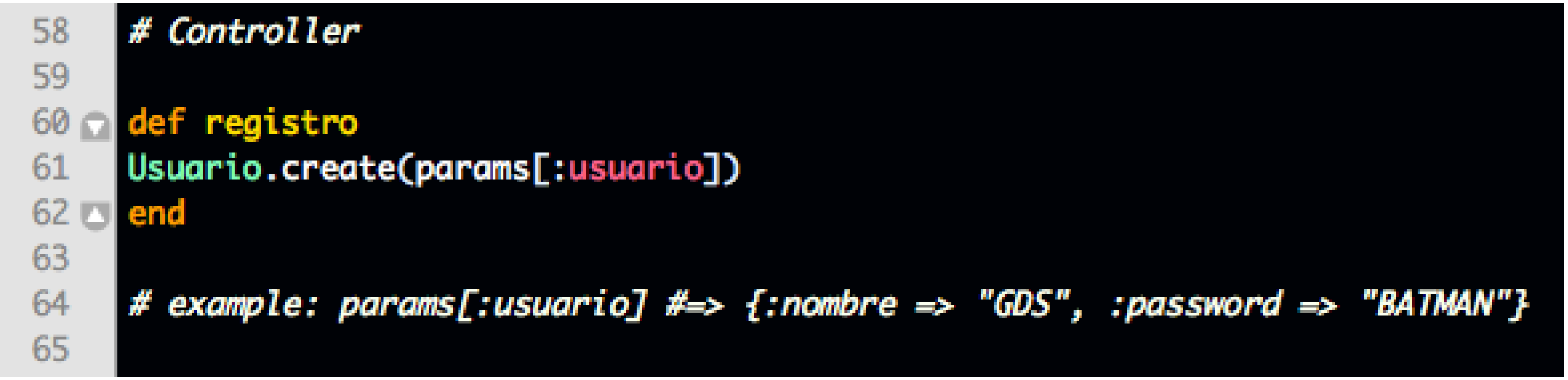
- Assign all the values received from a Form to model attributes
- Example: User sign-up process



```
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table :usuarios do |t| (
    t.string :nombre
    t.string :password
    t.string :rol, :default => "user"
    t.integer :aprobado, :default => 0
  end
  def self.down
    drop_table :usuarios
  end
end
```

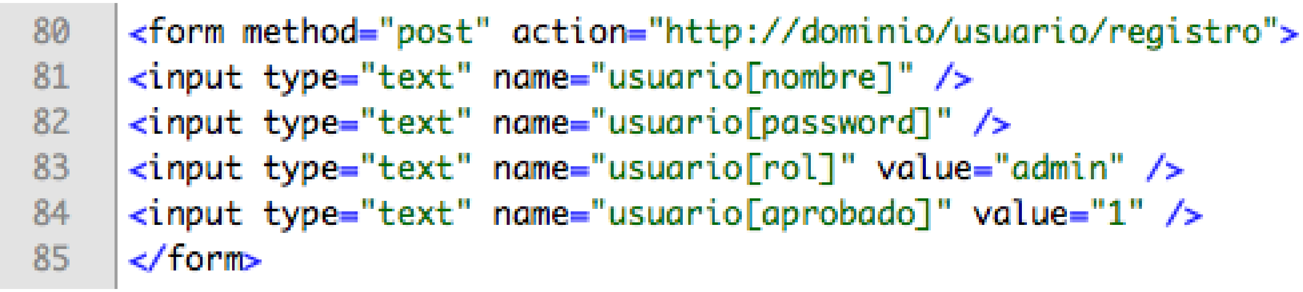
AUDITING RAILS APPLICATIONS

Vulnerabilities: Mass Assignment



```
# Controller

def registro
Usuario.create(params[:usuario])
end

# example: params[:usuario] #=> {:nombre => "GDS", :password => "BATMAN"}

```

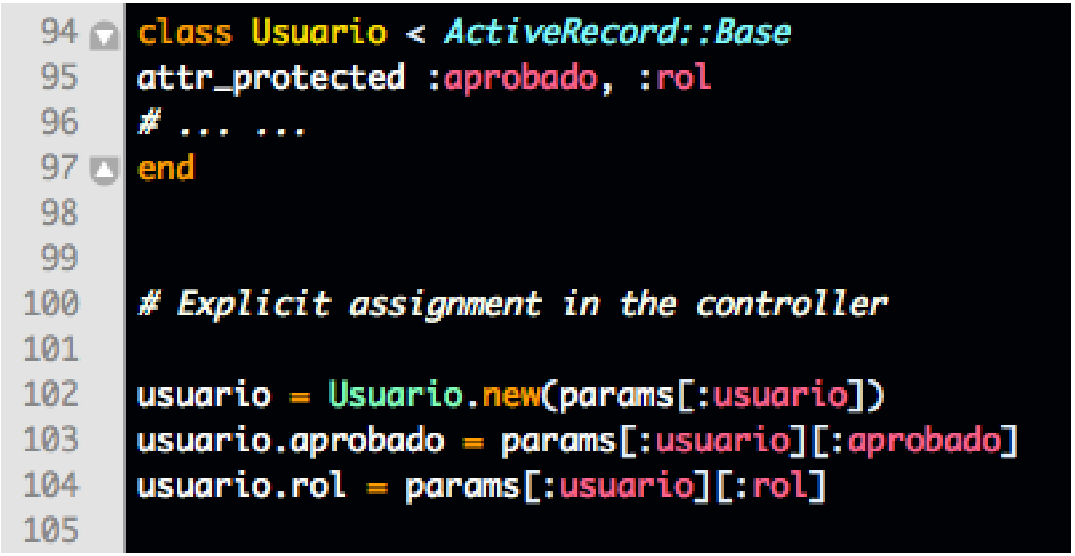
- What if …



```
<form method="post" action="http://dominio/usuario/registro">
<input type="text" name="usuario[nombre]" />
<input type="text" name="usuario[password]" />
<input type="text" name="usuario[rol]" value="admin" />
<input type="text" name="usuario[aprobado]" value="1" />
</form>
```

Vulnerabilities: Mass Assignment

- **REMEDIATION:**
  - Use attr_protected or attr_accessible



```
class Usuario < ActiveRecord::Base
attr_protected :aprobado, :rol
# ... ...
end


# Explicit assignment in the controller

usuario = Usuario.new(params[:usuario])
usuario.aprobado = params[:usuario][:aprobado]
usuario.rol = params[:usuario][:rol]

```

AUDITING RAILS APPLICATIONS

Vulnerabilities: Cross Site Scripting (XSS)

<script>alert(‘Hello:I am not just a popup’)</script>

- Formatting Allowed?
  - Use HTML and remove unwanted tags and attributes
- Earlier versions of Rails:
  - Blacklist approach for the strip_tags(), strip_links() and sanitize() helpers.
  - Injection was possible:

```
strip_tags("some<<b>script>alert('hello')<</b>/script>")
```

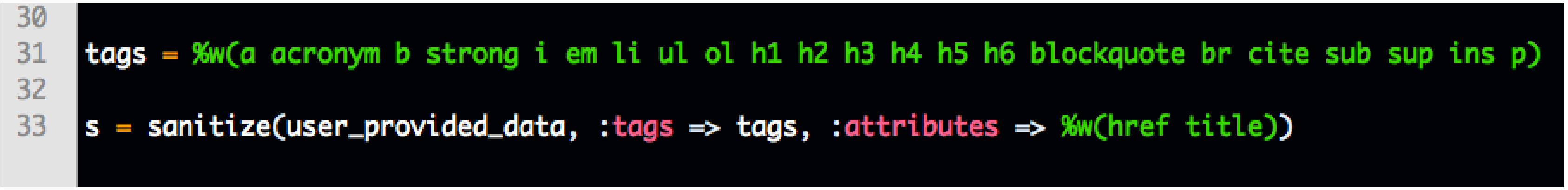
Vulnerabilities: Cross Site Scripting (XSS)

- Updated Rails 2 **sanitize()** helper
  - Removes protocols like “javascript:”
  - Filters HTML nodes and attributes
  - Handles unicode/ascii/hex hacks

- Second step to protect against xss:
  - Rails **h()** helper to HTML escape user input (easy to forget)
  - **escape_javascript()**
  - **safeERB** plugin. Raises an exception whenever a tainted string is not escaped
  - rails_xss plugin (Rails 2.3)

Vulnerabilities: Cross Site Scripting (XSS)

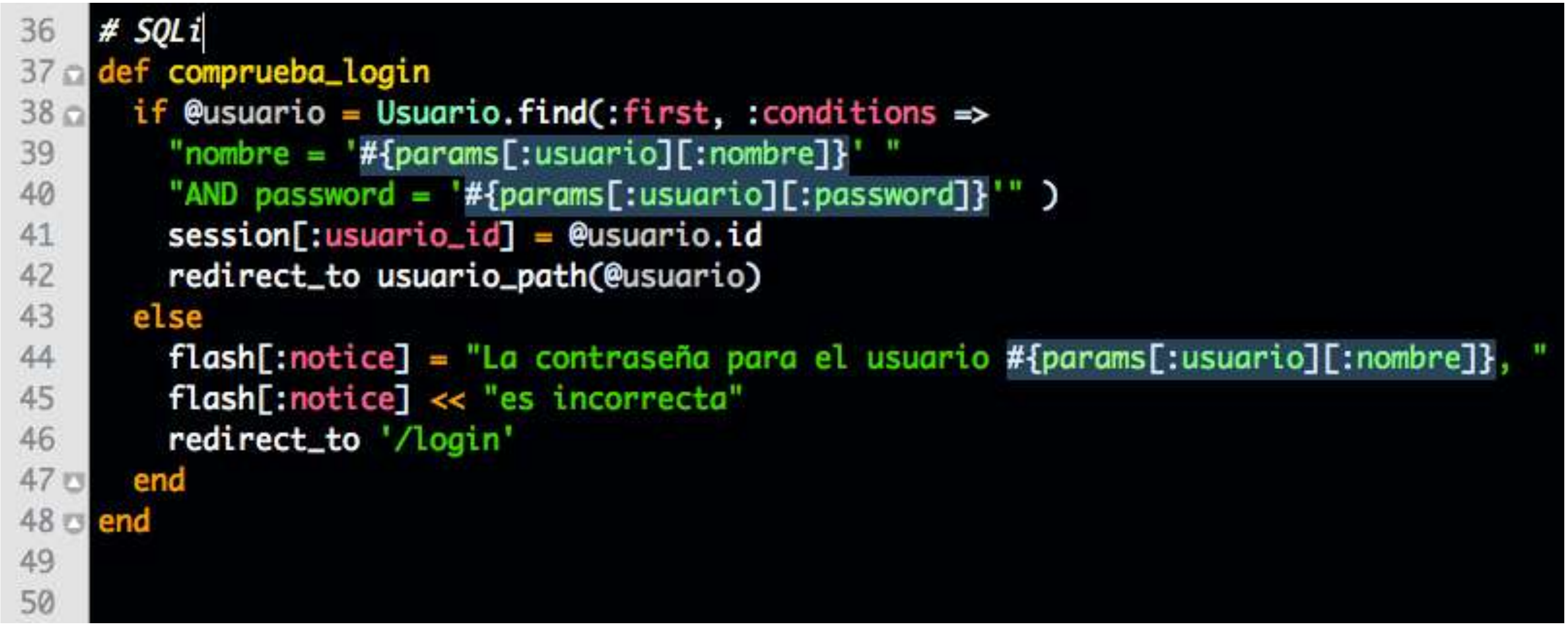
- Sanitize method:
  - Whitelisting (since Rails 2)



```
tags = %w(a acronym b strong i em li ul ol h1 h2 h3 h4 h5 h6 blockquote br cite sub sup ins p)

s = sanitize(user_provided_data, :tags => tags, :attributes => %w(href title))
```

- **Rails 3:**
  - Strings inside views are “automagically” scaped
  - Tainted strings? --> Call **"tainted text".html_safe**
  - Show the string as it is? **raw("I am tainted, you know ...”)**
  - XSS protection based on rails_xss plugin

Vulnerabilities: SQL Injection



```
# SQLi
def comprueba_login
  if @usuario = Usuario.find(:first, :conditions =>
    "nombre = '#{params[:usuario][:nombre]}' "
    "AND password = '#{params[:usuario][:password]}'" )
    session[:usuario_id] = @usuario.id
    redirect_to usuario_path(@usuario)
  else
    flash[:notice] = "La contraseña para el usuario #{params[:usuario][:nombre]}, "
    flash[:notice] << "es incorrecta"
    redirect_to '/login'
  end
end
```

AUDITING RAILS APPLICATIONS

Vulnerabilities: SQL Injection

- SELECT * FROM usuarios WHERE (nombre = '' AND password = '') LIMIT 1
- **INPUT:** something ' OR 'a'='a
- SELECT * FROM usuarios WHERE (nombre = 'GDS' AND password = 'something' OR 'a' = 'a') LIMIT 1

Vulnerabilities: SQL Injection

- **The right way:**
  - Use the methods find_(id) or dynamic methods such as: find_by_something(something)
  - Use find conditions with named bind variables:

```
Usuario.find(:first, :conditions => ["nombre = ? AND password = ?",
    nombre_usuario, clave])
```

```
Usuario.find(:first, :conditions => {:nombre => nombre_usuario,
    :password => clave})
```

- If using **connection.execute()** or **Model.find_by_sql()** custom filtering needs to be implemented

AUDITING RAILS APPLICATIONS

Vulnerabilities: Cross Site Request Forgery (CSRF)

- Is the security token active in the controller?
  - **protect_from_forgery :secret => "123456789012345678901234567890"**
- This does not check requests to XML APIs
- Restrict specific actions to specific HTTP methods:

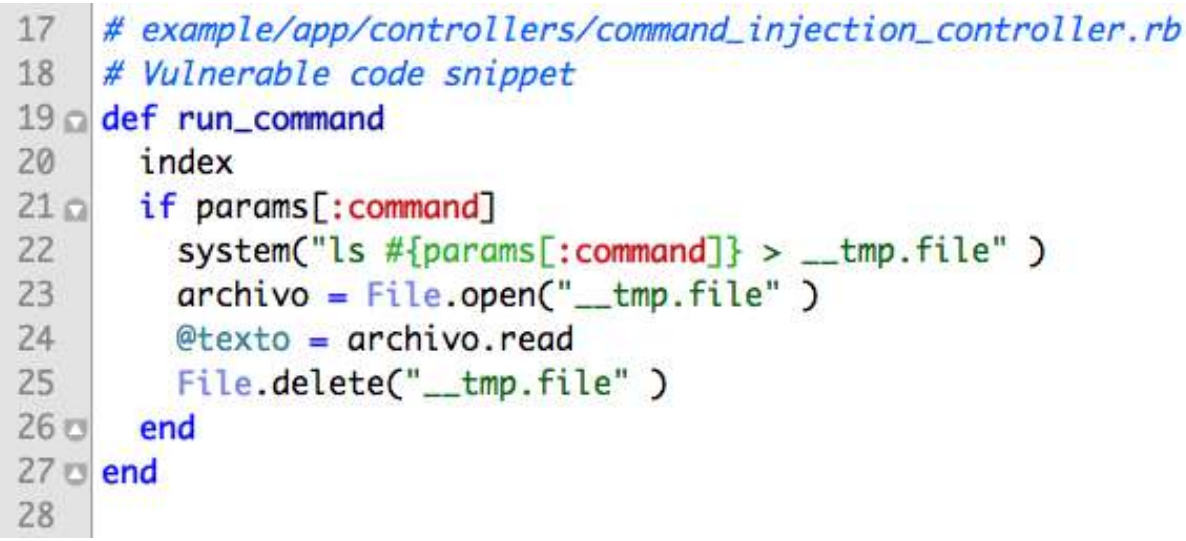
verify :method => :delete, :only => [:destroy], :redirect_to => {:action => :denegar}

```
<img src="http://dominio/projecto/1/destroy">
```

Vulnerabilities: Command Execution

- Ruby command execution:
  - exec(command)
  - system(command)
  - syscall(command)
  - \`command\`

system(command, parameters)

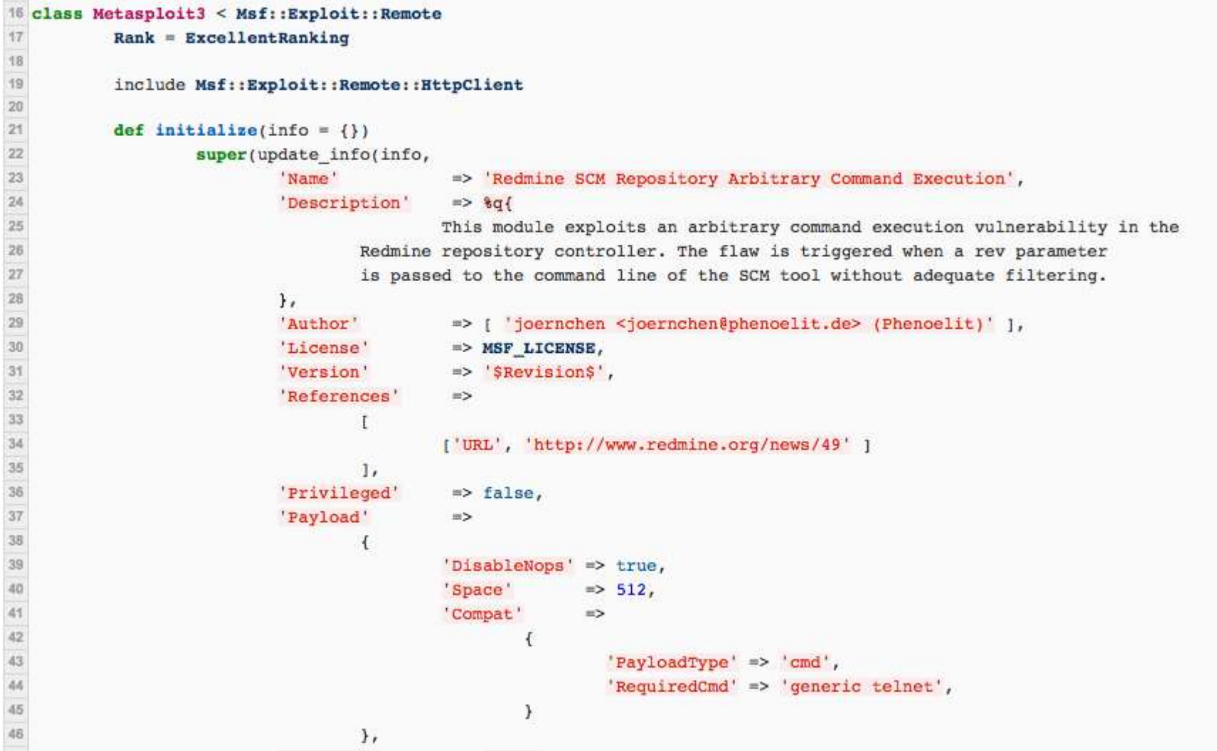


```
# example/app/controllers/command_injection_controller.rb
# Vulnerable code snippet
def run_command
  index
  if params[:command]
    system("ls #{params[:command]} > __tmp.file" )
    archivo = File.open("__tmp.file" )
    @texto = archivo.read
    File.delete("__tmp.file" )
  end
end
```

AUDITING RAILS APPLICATIONS

Vulnerabilities: Command Execution

- **Redmine SCM Repository Arbitrary Command Execution:**
- http://redminehost/projects/$project/repository/diff/?rev=`cmd`



```
class Metasploit3 < Msf::Exploit::Remote
        Rank = ExcellentRanking

        include Msf::Exploit::Remote::HttpClient

        def initialize(info = {})
                super(update_info(info,
                        'Name'           => 'Redmine SCM Repository Arbitrary Command Execution',
                        'Description'    => %q{
                                This module exploits an arbitrary command execution vulnerability in the
                        Redmine repository controller. The flaw is triggered when a rev parameter
                        is passed to the command line of the SCM tool without adequate filtering.
                        },
                        'Author'         => [ 'joernchen <joernchen@phenoelit.de> (Phenoelit)' ],
                        'License'        => MSF_LICENSE,
                        'Version'        => '$Revision$',
                        'References'     =>
                                [
                                        ['URL', 'http://www.redmine.org/news/49' ]
                                ],
                        'Privileged'     => false,
                        'Payload'        =>
                                {
                                        'DisableNops' => true,
                                        'Space'       => 512,
                                        'Compat'      =>
                                                {
                                                        'PayloadType' => 'cmd',
                                                        'RequiredCmd' => 'generic telnet',
                                                }
                                },
```

AUDITING RAILS APPLICATIONS

Checklist (Sort of)

- Search eRB files for <%= if its user input ensure it is HTML escaped
- Secure Access: check controllers and public actions
- Search for "forgery" make sure that config.action_controller.allow_forgery_protection = false is only disabled in test config
- Are passwords saved as clear-text in the db?, are being logged? **filter_parameter_logging**

AUDITING RAILS APPLICATIONS

Checklist (Sort of)

- Ensure private data is not stored in cookies
- Appropriate use of attr_accessible/attr_protected
- Is the application using validations inside models to prevent bad input?
- Are non-action controller methods private?
- Check for params[:id] usage
- Gems are up to date for latest security patches (rails security mailing list)
- Word search for "find", "first", and "all" "sql"
- Check for mass assignment

AUDITING RAILS APPLICATIONS

Tools: Brakeman

- **Static analysis security scanner for Ruby on Rails**
  - www.brakemanscanner.org
- Vulnerabilities Detected:
  - Cross site scripting
  - SQL injection
  - Command injection
  - Unprotected redirects
  - Unsafe file access
  - Version-specific security issues
  - Unrestricted mass assignment
  - Dangerous use of eval() Default routes
  - Insufficient model validation

AUDITING RAILS APPLICATIONS

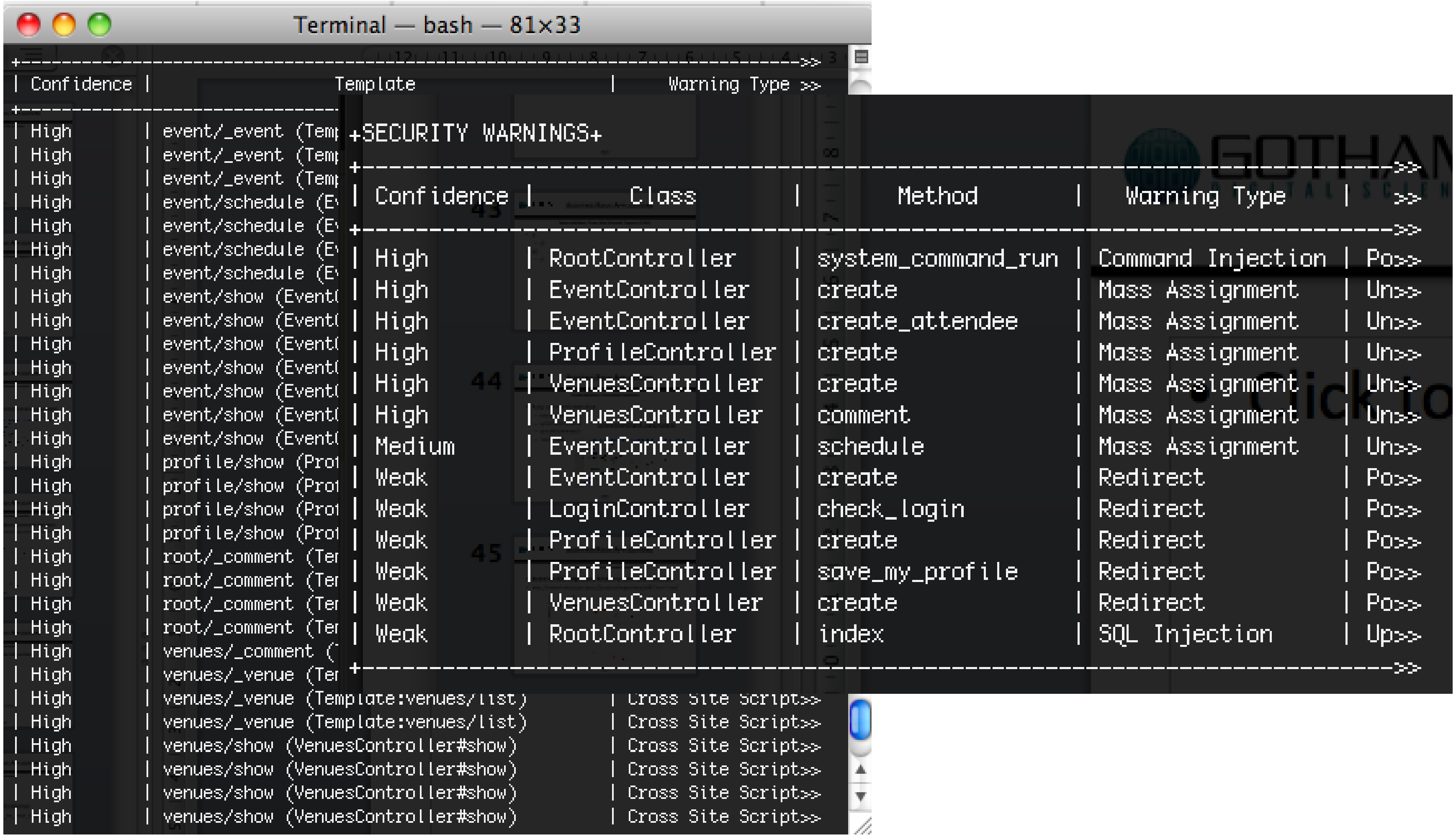
Tools: Brakeman

- **Using Brakeman**

```
gem install brakeman
brakeman –p /path_to_your_rails_app
```

AUDITING RAILS APPLICATIONS

Tools: Brakeman



```
+------------------------------------------------------------------------->>
| Confidence |                   Template                   |   Warning Type >>
+----------------------------------
| High       | event/_event (Temp
| High       | event/_event (Temp
| High       | event/_event (Temp
| High       | event/schedule (E
| High       | event/schedule (E
| High       | event/schedule (E
| High       | event/schedule (E
| High       | event/show (Event
| High       | event/show (Event
| High       | event/show (Event
| High       | event/show (Event
| High       | event/show (Event
| High       | event/show (Event
| High       | event/show (Event
| High       | profile/show (Prot
| High       | profile/show (Prot
| High       | profile/show (Prot
| High       | profile/show (Prot
| High       | root/_comment (Ter
| High       | root/_comment (Ter
| High       | root/_comment (Ter
| High       | root/_comment (Ter
| High       | venues/_comment (
| High       | venues/_venue (Ter
| High       | venues/_venue (Template:venues/list)           | Cross Site Script>>
| High       | venues/_venue (Template:venues/list)           | Cross Site Script>>
| High       | venues/show (VenuesController#show)            | Cross Site Script>>
| High       | venues/show (VenuesController#show)            | Cross Site Script>>
| High       | venues/show (VenuesController#show)            | Cross Site Script>>
| High       | venues/show (VenuesController#show)            | Cross Site Script>>
```

```
+SECURITY WARNINGS+

+---------------------------------------------------------------------------->>
| Confidence |        Class        |        Method        |    Warning Type    | >>
+---------------------------------------------------------------------------->>
| High       | RootController      | system_command_run   | Command Injection  | Po>>
| High       | EventController     | create               | Mass Assignment    | Un>>
| High       | EventController     | create_attendee      | Mass Assignment    | Un>>
| High       | ProfileController   | create               | Mass Assignment    | Un>>
| High       | VenuesController    | create               | Mass Assignment    | Un>>
| High       | VenuesController    | comment              | Mass Assignment    | Un>>
| Medium     | EventController     | schedule             | Mass Assignment    | Un>>
| Weak       | EventController     | create               | Redirect           | Po>>
| Weak       | LoginController     | check_login          | Redirect           | Po>>
| Weak       | ProfileController   | create               | Redirect           | Po>>
| Weak       | ProfileController   | save_my_profile      | Redirect           | Po>>
| Weak       | VenuesController    | create               | Redirect           | Po>>
| Weak       | RootController      | index                | SQL Injection      | Up>>
+---------------------------------------------------------------------------->>
```

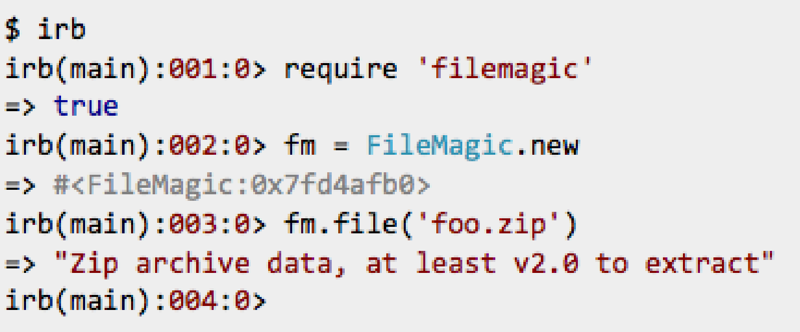
Tips – Gems – Plugins

BUILDING SECURE APPLICATIONS

SECURITY GOODNESS WITH RUBY ON RAILS

Recommendations: File uploads

- Analyze the files with Antivirus
- Random name. Save outside DocumentRoot
- **Avoid potential DOS** (asyncronous tasks). Resque to the rescue!
- Validate the MIME type
- Ruby binding to libmagic (ruby-filemagic)
- shared-mime-info gem. Not recognized? Modify MIME.check(file)
- Serving the files later? send_file :disposition => 'attachment’



```
$ irb
irb(main):001:0> require 'filemagic'
=> true
irb(main):002:0> fm = FileMagic.new
=> #<FileMagic:0x7fd4afb0>
irb(main):003:0> fm.file('foo.zip')
=> "Zip archive data, at least v2.0 to extract"
irb(main):004:0>
```

Tips: Authentication

- Popular authentication plugins:
  - RestfulAuthentication
  - Authlogic
- Popular SSO systems:
  - OpenID
  - CAS
  - Kerberos
  - GSS-API
  - SPNEGO
  - OAuth (gem install oauth)
  - LDAP (gem install ruby-net-ldap)

Tips: Authorization

- **Mandatory access control (MAC):**
  - Grants access based on the sensitivity of the information (i.e., clearance)
  - Example: Government information classification, such as Secret or Top Secret
- **Discretionary access control (DAC):**
  - Grants access to objects based on the identity of subjects and/or groups to which they belong.
  - Example: Windows and Unix file permissions
- **Role-based access control (RBAC):**
  - Access to actions is controlled through permission based on role assignments, not at the level of individual data objects.
  - Example: Active Directory

BUILDING SECURE APPLICATIONS

Tips: Authorization

- Simple Solutions: **role_requirement** (http://code.google.com/p/rolerequirement/).
- Complex Scenarios: **DeclarativeAuthorization** plugin (RBAC) (http://github.com/stffn/declarative_authorization)

- Other interesting plugins:
- **ActiveRbac** (http://active-rbac.rubyforge.org/).
- **ModelSecurity** (http://perens.com/FreeSoftware/ModelSecurity/).

BUILDING SECURE APPLICATIONS

Tips: Admin Interface & good practices

- Isolate administrative interface (subdomain, authentication, restricted)
- Check request.remote_ip
- Digital Certificates
- **Two factor auth** (ROTP - The Ruby One Time Password Library https://github.com/mdp/rotp)
- Alerts (invalid logins, suspicious activity)
- Mandatory use of secure protocols (ActionController::Base.session_options[:session_secure] = true)
- Cookies with httponly and secure flags
- Deployment:
  - Passwords inside database.yml
  - Subversion files
  - Test files

THANKS FOR COMING! ANY QUESTIONS?

dpelaez@gdssecurity.com