INSECURE EXPECTATIONS

MATT KONDA
@MKONDA
JEMURAI.COM
INTRODUCTION

MATT KONDA
@MKONDA
MKONDA@JEMURAI.COM
THANKS TO FAMILY!
cucumber --name "person is restricted from putting input into a field that will be executed by the system"
def destroy
  @project = Project.find(params[:id])

  name = @project.name
  `rm /tmp/#{name}.log`

  @project.destroy

  respond_to do |format|
    format.html { redirect_to projects_url }
    format.json { head :no_content }
  end
end

What if @project.name is :
"; cat /etc/passwd > public/passwd.html;"
HOW MANY PEOPLE HERE

WRITE TESTS?
HOW MANY PEOPLE HERE

USE TDD?
HOW MANY PEOPLE HERE

USE BDD?
How many people here know of OWASP?
How many people here currently write security tests?
INSECURE
EXPECTATIONS
rspec
Trading error worth trillions disrupts Swedish market

-6 = 64.1 Trillion
require 'spec_helper'

describe User do
  before(:each) do
    @user = User.new
    @user.email = "hi@hi.com"
  end

  it "should not allow short passwords" do
    @user.password = "hi1B"
    @user.save
    @user.errors.should have(1).messages
    @user.errors.messages[:password].should eql ["is too short (minimum is 8 characters)"
  end

  it "should not allow passwords without a digit" do
    @user.password = "highthere"
    @user.save
    @user.errors.should have(1).messages
    @user.errors.messages[:password].should eql ["must include at least one lowercase letter, one uppercase
  end

  it "should not allow passwords without an alpha" do
    @user.password = "32434234324"
    @user.save
    @user.errors.should have(1).messages
    @user.errors.messages[:password].should eql ["must include at least one lowercase letter, one uppercase
  end

  it "should accept complex passwords with a lower, upper and digit" do
    @user.password = "Passw0rd!"
    @user.save
    @user.errors.should be_empty
  end
end
FEATURE
SCENARIO

GIVEN
WHEN
THEN
Feature: person is restricted from accessing project they do not own

Scenario: person accesses a project that is not theirs

Given a new project created by a user
When a different person attempts to access the project
Then the system should prevent access
cucumber --name "person is restricted from accessing project they do not own"
Given(/^a new project created by a user$/) do
  uuid = SecureRandom.uuid
  @user1 = "fb_user_1_#{uuid}@jemurai.com"
  register_as_user(@user1, "password")
  new_project("Insecure Direct Object Reference #{uuid}",
              "Forceful Browsing Desc")
  @url = current_url
end

When(/^a different person attempts to access the project$/) do
  logout(@user1)
  uuid = SecureRandom.uuid
  @user2 = "fb_user_2_#{uuid}@jemurai.com"
  register_as_user(@user2, "password")
end

Then(/^the system should prevent access$/) do
  visit @url
  expect(page).not_to have_content "Forceful Browsing Desc"
end
INTRODUCING: TRIAGE

https://github.com/Jemurai/triage
def index
  email = current_user.email
  conditions = "owner LIKE '#{email}'"
  if params[:name]
    conditions = "name like '#{params[:name]}" + conditions
  end
  @projects = Project.find(:all, :conditions=>conditions)

  respond_to do |format|
    format.html # index.html.erb
    format.json { render json: @projects }
  end
end

SELECT "projects".* FROM "projects"
WHERE (name like 'A') or 1=1 -- owner LIKE 'test@test.com'
INTRODUCING: SWTFT
SECURITY WEB TESTING FRAMEWORK
SECURITY WTF
require 'cucumber/formatter/unicode'

require 'capybara/cucumber'
require 'securerandom'

Capybara.default_driver = :selenium

Capybara.app_host = 'http://triaze-secure:3003/

#Capybara.app_host = 'http://triaze-insecure:3000/'
module TriageDriver

  def register_as_user(username, password)
    visit 'users/sign_up'
    fill_in "user[email]", :with => username
    fill_in "user[password]", :with => password
    fill_in "user[password_confirmation]", :with => password
    click_button "Create My Account"
  end

  def logout(username)
    click_link "Logout #{username}"
  end

  def login_as_user(username, password)
    visit '/users/sign_in'
    fill_in "user[email]", :with => username
    fill_in "user[password]", :with => password
    click_button "Sign in"
  end

  def access_project(id)
    visit '/projects/' + id
  end

  def new_project(name, description = nil, priority = 3, rank = 3, tier = 3, verified = false, rich_description = nil)
    visit '/projects/
    click_button "New Project"
    fill_in "project[name]", :with => name if name
    fill_in "project[description]", :with => description if description
    fill_in "project[priority]", :with => priority if priority
    fill_in "project[rank]", :with => rank if rank
    fill_in "project[tier]", :with => tier if tier
    # fill_in "project[verified]", :with => verified if verified
    fill_in "project[rich_description]", :with => rich_description if rich_description
    click_button "Create Project"
  end

end

World(TriageDriver)
Feature: user is prevented from putting XSS in project form fields

A user wants to be sure that others users can’t put XSS in the projects pages
in order to ensure that their sessions and information are safe.

@javascript

Scenario Outline: xss attempt
  Given the field is "<fieldname>"
  When the value is "<value>"
  Then the field result should be "<result>"

Scenarios: xss in fields
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>field name</td>
<td>value</td>
<td>result</td>
<td></td>
</tr>
<tr>
<td>project[name]</td>
<td>ProjectName</td>
<td>noxss</td>
<td></td>
</tr>
<tr>
<td>project[name]</td>
<td>ProjectName &lt;script&gt;alert('project[name]-&gt;xss');&lt;/script&gt;</td>
<td>xss</td>
<td></td>
</tr>
<tr>
<td>project[description]</td>
<td>ProjectDescription &lt;script&gt;alert('project[description]-&gt;xss');&lt;/script&gt;</td>
<td>noxss</td>
<td></td>
</tr>
</tbody>
</table>
new_project("XSS Name #{@field} #{uniq}", "XSS Desc #{@field}" + uniq)
click_link 'Edit'
fill_in @field, :with => @value
click_button "Update Project"
if @result == "xss"
    # This should have xss in it...did it stick?
    alerted = false
    begin
        page.driver.browser.switch_to.alert.accept
        alerted = true
    rescue
    end
    if alerted
        fail("XSS Used to create Popup in #{@field} with #{@value}")
    else
        puts "Good news, no xss where expected."
    end
else
    expect(page).to have_content @value
end
cucumber --name "user is prevented from putting XSS in project form fields"
Jemurai Triage
Making it easy to manage application security.

Sign Up
Email
d-b4a6-362961a2cc91@jemurai.com
Password
Password confirmation
Your Password
Create My Account

Jemurai, LLC © 2013

---

```javascript
Scenario Outline: xss attempt
| features/xss/project_xss.feature:6 |
| features/step_definitions/xss_step_definitions/project_xss_steps.rb:1 |
| features/step_definitions/xss_step_definitions/project_xss_steps.rb:5 |
| Then the field result should be "result" |

Scenarios: xss in fields
<table>
<thead>
<tr>
<th>fieldname</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Given the field is "(.*)" do |arg1|
  @field = arg1
end

When the value is "(.*)" do |arg1|
  @value = arg1
end

Then the field result should be "(.*)" do |arg1|
  @result = arg1
  uniq = Time.now.to_s
  run = SecureRandom.uuid
  user = "test+#{run}@jemurai.com"
  register_as_user(user, "password")
  # logout(user)
  # login_as_user(user, 'password')
  new_project("XSS Name #{@field} #{uniq}", "XSS Desc #{@field}+ uniq")
  click_link 'Edit'
  fill_in @field, :with => @value
  click_button "Update Project"
  if @result == "xss"
    # This should have xss in it...did it stick?
    alerted = false
    begin
      page.driver.browser.switch_to.alert.accept  # For now assume any XSS has an alert.
      alerted = true
    rescue
    end
    if alerted
      fail("XSS Used to create Popup in #{@field} with #{@value}")
    else
      puts "Good news, no xss where expected."
    end
  else
    puts "No dialog..."
  end
  expect(page).to have_content @value
end
Tests in App

Rails Application

rspec / cucumber
TESTS OUT OF APP

Rails Application: Triage

Cucumber | SWTF
Tests out of app

Rails Application: Triage (Insecure)

Rails Application: Triage (Secure)

Cucumber | SWTF
Means they can be easily adapted to test different apps.
cucumber --name "user is protected from malicious content and having their page framed"
Feature: user is protected from malicious content and having their page framed

A user wants to be sure that effective browser protections are enabled
in order to ensure that their information is safe.

@javascript
Scenario Outline: check for secure headers attempt
Given a new project created by a user
And the page is "<page>"
When the header is "<header>"
Then the header value should be "<result>"

Scenarios: headers in pages

<table>
<thead>
<tr>
<th>page</th>
<th>header</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>projects/</td>
<td>X-Frame-Options</td>
<td>DENY</td>
</tr>
<tr>
<td>projects/</td>
<td>X-XSS-Protection</td>
<td>1</td>
</tr>
</tbody>
</table>
cookies = Capybara.current_session.driver.browser.manage.all_cookies
csrf_token = Capybara.current_session.driver.browser.find_element(:xpath, "//meta[@name='csrf-token']").attribute('content');

# Switch mode to net::http
uri = URI.parse(url)
http = Net::HTTP.new(uri.host, uri.port)
http.verify_mode = OpenSSL::SSL::VERIFY_NONE
request = Net::HTTP::Post.new(uri.request_uri)
request['Cookie'] = cookies
request.set_form_data(
  "_method" => "put",
  "authenticity_token" => "#{@csrf_token}",
  "project[name]" => "header updated and verified",
  "commit" => "Update Project"
)
response = http.request(request)

...

if response[@header] == @result
  #pass
else
  fail("Header #{@header} not set to #{@result} as expected. Instead was #{@response[@header]}.")
end
TAKE A VULNERABLE PROJECT
WRITE TESTS THAT ILLUSTRATE THE SECURITY ISSUES
TRY TO ILLUSTRATE HOW EASY IT WOULD BE TO WRITE SECURITY TESTS
IN LANGUAGE EVERYONE CAN UNDERSTAND
WHY IS APPLICATION SCANNING SO HARD?
WHAT IF THE DEV WRITING THE CODE WERE TESTING SECURITY CASES ALONG THE WAY?

MUCH SMARTER.
EXPLORATORY TESTING
QUIZ

• User should not be able to set fields not shown in the form
QUIZ

• USER SHOULD NOT BE ABLE TO SUBMIT FORMS IN ANOTHER SESSION
ARE STAKEHOLDERS ASKING FOR SECURITY?
but if you ask them about these features they might want them
CURRENT TESTS

- INJECTION / SQL INJECTION
- CROSS SITE SCRIPTING
- MASS ASSIGNMENT
- CROSS SITE REQUEST FORGERY
- SECURE HEADERS
- SENSITIVE DATA EXPOSURE (SESSION COOKIE)
cucumber --name "users favorite album is in cookie"
cucumber features/xss/project_xss.feature:6 # Scenario: xss attempt

3 scenarios (1 failed, 2 passed)
9 steps (1 failed, 8 passed)

jemural:swf mkondo$ cucumber --name "user is protected from malicious content and having their page framed"
Feature: user is protected from malicious content and having their page framed
A user wants to be sure that effective browser protections are enabled in order to ensure that their information is safe.

@javascript
Scenario Outline: check for secure headers attempt # features/headers/security_headers.feature:6
Given a new project created by a user # features/step_definitions/InsecureDirectObjectReference/insecure_direct_object_reference_step.rb:1
And the page is "<pages>
When the header is "<headers>
Then the header value should be "<result>
Scenarios: headers in pages
<table>
<thead>
<tr>
<th>page</th>
<th>header</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>projects/</td>
<td>-</td>
<td>X-Frame-Options</td>
</tr>
</tbody>
</table>
Header X-Frame-Options not set to DENY as expected. Instead was . (RuntimeError)

Failing Scenarios:
cucumber features/headers/security_headers.feature:6 # Scenario: check for secure headers attempt
cucumber features/headers/security_headers.feature:6 # Scenario: check for secure headers attempt

2 scenarios (2 failed)
8 steps (2 failed, 6 passed)
0m57.068s
jemural:swf mkondo$ cucumber --name "users favorite album is in cookie"
When(/\^the accesses the dashboard$/) do
  visit "/
end

Then(/\^the session cookie should not contain sensitive information$/) do
  cookies = Capybara.current_session.driver.browser.manage.all_cookies

  cookie = cookies[0]
  detail = cookie[:value]
  # puts "Detail is #{detail}"
  puts "Cookie is #{cookie}"

  decoded = ""
  begin
    decoded = Marshal.load(Base64.decode64(detail))
  rescue
    decoded = ""
  end

  # puts decoded
  expect(decoded).not_to have_content ""CTF_FLAG"="2112"

  # In prod this should be true.
  # expect(cookie).to have_content 'HttpOnly' # This is not coming through for some reason. Grr

  # In prod these should also be applicable.
  # expect(cookie).to have_content ':secure=>true'

  expect(cookie).not_to have_content ':expires=>nil'
end
Simplified Steps

- **Injection:** Inject commands into fields and detect functions being called.
- **XSS:** Inject scripts into fields and detect that alerts are thrown.
- **Mass Assignment:** Set raw form data with net::http and send it to see how the server responds.
- **CSRF:** Alter CSRF token and send otherwise valid request.
- **Headers:** Interact with system and verify that headers are being set.
- **Sensitive Data:** Open session cookie and inspect.
## OWASP Top 10

### A1-Injection
Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

### A2-Broken Authentication and Session Management
Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.

### A3-Cross-Site Scripting (XSS)
XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

### A4-Insecure Direct Object References
A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

### A5-Security Misconfiguration
Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.

### A6-Sensitive Data Exposure
Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

### A7-Missing Function Level Access Control
Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.

### A8-Cross-Site Request Forgery (CSRF)
A CSRF attack forces a logged-in victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

### A9-Using Components with Known Vulnerabilities
Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

### A10-Unvalidated Redirects and Forwards
Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.
A NATURAL EXTENSION IS TO MAKE IT EASY TO FUZZ FORMS
THERE IS NO
GOAL #1: PLATFORM FOR EDUCATING DEVELOPERS
GOAL #2: LANGUAGE FOR COMMUNICATING WITH BUSINESS OWNERS
GOAL #3: MECHANISM FOR MAKING IT EASIER TO IMPLEMENT TESTS.
HOW MANY PEOPLE HAVE HAD A PENETRATION TEST AGAINST THEIR APPLICATION?
INSTEAD OF A PDF, WHAT IF WE DELIVER FINDINGS WITH WORKING TESTS!
WHAT IF A DEVELOPER COULD FIX A SECURITY ISSUE BY MAKING THE TEST PASS.
WANDER
GOOD: ALMOST EVERY CLIENT ENGAGEMENT BENEFITS FROM OWASP
BAD: owasp
meetings I have been to are predominantly security people
HOW DO WE
“COMMUNICATE”
CHEAT SHEETS
How many people here have attended developer conferences this year?
How many people here! Commit to development projects?
COMMUNITY ORGANIZING
IDEAS:

GO TO DEV MEETUP
GO DEV CONFERENCE
CONTRIBUTE TO OSS
LISTEN
MORE IDEAS:

IDENTIFY TECHNOLOGY LEADERS AND APPROACH THEM
MAKE DEVELOPER FRIENDS
MAKE DEVELOPER FRIENDS
MORE IDEAS:

FORM AN OUTREACH SUBCOMMITTEE
MORE IDEAS:

DE-"CRIMINALIZE"
APPLICATION
SECURITY
IGNORANCE
MORE IDEAS:

INVITE DEVELOPERS TO SPEAK
MORE IDEAS:

FIND ACTIVITIES THAT DEVELOPERS CAN PARTICIPATE IN AT MEETINGS
MORE IDEAS:

EMPHASIZE DEVELOPER CONTRIBUTIONS TO OWASP SITE
MORE IDEAS:
GET DEVELOPERS ON THE OWASP BOARD
MORE IDEAS:

???
BASICALLY, I WANT TO SEE OWASP TRY TO BUILD COMMUNITY ORGANIZING WITH DEVELOPERS INTO A MODEL THAT CAN BE REPEATED
THANKS!

Justin Collins @presidentbeef
Jeff Jarmoc @jjarmoc
Ben Toews @mastahyeti
Neil Matatall @ndm
Aaron Bedra @abedra
Jon Claudius @claudijd
Chris Oliver @excid3
Chris Hildebrand @ckhrysze
Jon Rose
Brett Hardin @miscsecurity
Elizabeth Hendrickson @testobsessed
REFERENCES

- https://github.com/Jemurai/triage
- https://bitbucket.org/mkonda/swtf/
- http://speakerdeck.com/mkonda
- http://brakemanscanner.org
- http://rails-sqli.org
- https://github.com/twitter/secureheaders
- https://www.owasp.org/index.php/Ruby_on_Rails_Cheatsheet
THANKS!
FEATURES

• PERSON IS RESTRICTED FROM PUTTING INPUT INTO A FIELD THAT WILL BE EXECUTED BY THE SYSTEM

• USER IS PREVENTED FROM PUTTING XSS IN PROJECT FORM FIELDS

• USER SHOULD NOT BE ABLE TO SET FIELDS NOT SHOWN IN THE FORM

• USER SHOULD NOT BE ABLE TO SUBMIT FORMS IN ANOTHER'S SESSION

• USER IS PROTECTED FROM MALICIOUS CONTENT AND HAVING THEIR PAGE FRAMED

• USERS FAVORITE ALBUM IS IN COOKIE