

Security Testing with Selenium

Vidar Kongsli

Montréal, October 25th, 2007



Versjon 1.0

- Vidar Kongsli

- ✓ System architect & developer
- ✓ Head of security group

- Bekk Consulting

- ✓ Technology and Management Consulting
- ✓ Based in Oslo, Norway
- ✓ Focus on agile methodologies

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

■ Processes

- ✓ A: System design and implementation grows incrementally
- ✓ S: Security review typically after design and after implementation

■ People

- ✓ A: Focused team, co-located in same office space
- ✓ S: Security reviews should be done by externals, not part of the development team

■ Documentation

- ✓ A: Lean. More effective communication within team preferred
- ✓ S: Security reviews based on system documentation

■ Model

- ✓ A: No modeling. Design emerges from test-driven development
- ✓ S: System, risk, threats, should be modeled

■ Leverage

- ✓ Techniques and tools of agile projects
- ✓ Common ownership
- ✓ Automated testing.

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

- What
 - ✓ Describes illegal or non-normative use of the system
- How
 - ✓ Derived from a user story or stories
 - ✓ Question: *"how can this functionality be misused?"*

- Misuse story:
 - ✓ "As a non-privileged anonymous user, I can create a new account with administrator privileges and use it"
- User story:
 - ✓ "As an anonymous user, I can create a user account"
- Flaw:
 - ✓ Mass assignment allows a user to inject admin flag when submitting user account information.

- Misuse story:

- ✓ "As a logged in user, I can insert JavaScript in my post which will be executed when another person reads my post"

- User story:

- ✓ "As a logged in user, I can post to my blog"

- Flaw:

- ✓ No meta character escaping or white listing of legal HTML tags in posts in place.

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

- Web testing
- JavaScript based
- Runs the application under test in an HTML frame
- Test cases
 - ✓ Can be recorded
 - ✓ Can be written in several languages

Selenium Functional Test Runner v0.8.2 [1727] - Mozilla Firefox

File Edit View History Bookmarks Tools Help del.jcio.us

http://localhost:3000/selenium/TestRunner.html?test=tests/normal

Suites: **Normal**

- Login.Simple login and logout
- Navigation.Navigate blogs
- Navigation.View frontpage
- Posting.Post an entry

Post an entry		
open	/selenium/setup?fixtures=all	
open	/	
deleteCookie	_OOPSLA2007_session_id	/
open	/login/login?original_url=/	
type	name	gandalfw
type	password	mellon
clickAndWait	commit	
clickAndWait	link=The life of a Wizard	
clickAndWait	link=Add entry	
assertTextPresent	New entry in 'The life of a Wizard'	
type	id=entry_title	Airport security

Selenium TestRunner

Execute Tests: [Play] [Stop] [Pause] [Refresh]

Fast [Slider] Slow

Highlight elements

Elapsed: 00:02

Tests **Commands**

4 run 24 passed

0 failed 0 failed

0 incomplete

Tools: [View DOM] [Show Log]

Create user | Log in

The life of a Wizard

Logged out

Thu Oct 18 17:21:47 +0200 2007

Airport security

When travelling from the Montreal-Trudeau airport today, I was not allowed to bring my staff as carry-on luggage...

Posted by Gandalf White at Thu Oct 18 17:21:47 +0200 2007

Sun Jan 14 17:12:02 +0100 2007

Harry Potter does not impress me

I have a much bigger stick than his little silly wand.

Posted by Gandalf White at Sun Jan 14 17:12:02 +0100 2007

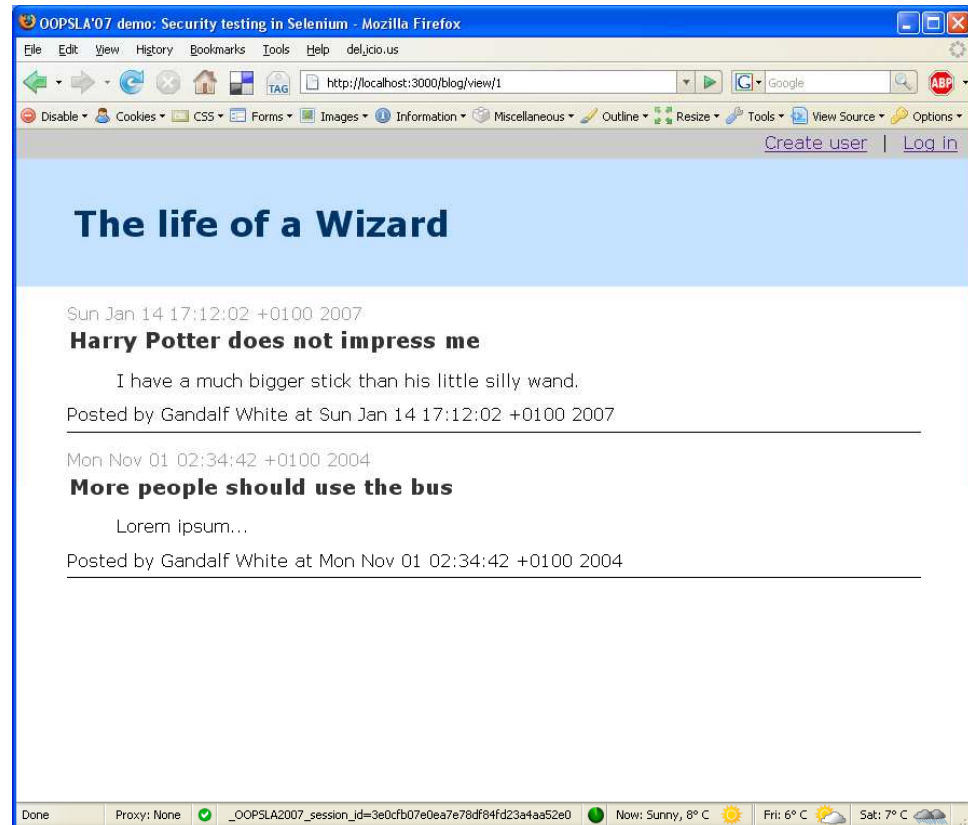
Mon Nov 01 02:34:42 +0100 2004

Done Proxy: None [OK] _OOPSLA2007_session_id=b77327b9cdc7be9dd7ffec4ed6c6007a Now: Sunny, 8° C Fri: 6° C Sat: 7° C

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

The demo application

- Simple weblog application
 - ✓ Register users
 - ✓ Create blogs
 - ✓ Write blog entries
- Written in Ruby on Rails
 - ✓ Quick development
 - ✓ Very good testability
 - ✓ Not always secure – lets you run with scissors



- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

■ Type 2 (Source: Wikipedia)

- ✓ Bob hosts a web site which allows users to post messages and other content to the site for later viewing by other members.
- ✓ Mallory notices that Bob's website is vulnerable to a type 2 XSS attack.
- ✓ Mallory posts a message, controversial in nature, which may encourage many other users of the site to view it.
- ✓ Upon merely viewing the posted message, site users' session cookies or other credentials could be taken and sent to Mallory's webserver without their knowledge.
- ✓ Later, Mallory logs in as other site users and posts messages on their behalf....

■ Test:

- ✓ Can I insert JavaScript code that is run when a victim views the page?

Post xss(misuse)		
open	/selenium/setup?fixtures=all	
open	/login/login?original_url=/	
type	name	gandalfw
type	password	mellon
clickAndWait	commit	
clickAndWait	link=The life of a Wizard	
clickAndWait	link=Add entry	
assertTextPresent	New entry in 'The life of a Wizard'	
type	id=entry_title	XSS check
type	id=entry_body	<script>document.title='Hackd by Saruman'</script>
clickAndWait	commit	
assertNotTitle	Hackd by Saruman	
clickAndWait	link=Log out	

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

- The following characteristics are common to CSRF:
 - ✓ Involve sites that rely on a user's identity
 - ✓ Exploit the site's trust in that identity
 - ✓ Trick the user's browser into sending HTTP requests to a target site
 - ✓ Involve HTTP requests that have side effects
 - ✓ (Source: Wikipedia)
- Scenario in our demo app:
 - ✓ Post a blog entry on behalf of another, logged in, user
- Countermeasures:
 - ✓ Preventing XSS is a good start
 - ✓ Including a secret, user specific "ticket" that is validated when the user submits the form

■ Test:

- ✓ Can we post a form (postback) without reading it first?

<i>Forge post(misuse)</i>		
open	/selenium/setup?fixtures=all	
open	/login/login?original_url=/	
type	name	gandalfw
type	password	mellon
clickAndWait	commit	
open	/test/csrf_add_entry.htm	
clickAndWait	commit	
assertTextNotPresent	New entry entitled 'Faked title' created	

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

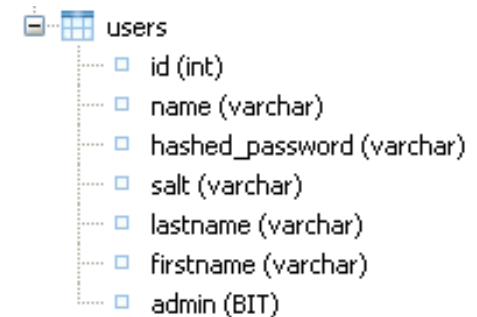
■ Test:

- ✓ Can I register a user that has administrator privileges?

Set user as admin(misuse)		
open	/selenium/setup?fixtures=all	
open	/login/add_user	
addHiddenField	user[admin]	true
type	user_firstname	Dr
type	user_lastname	Evil
type	user_name	drevil
type	user_password	numbertwo
type	user_password_confirmation	numbertwo
clickAndWait	commit	
type	password	numbertwo
clickAndWait	commit	
open	/login/list_users	
assertTextPresent	not authorized	

Why did this work

- Database table contains a flag which tells if the user is an administrator
- Rails uses "mass assignment" where it automatically maps form parameters into database table by name
- For the test, I added an action in Selenium that inserts a hidden field into the form
- Test inserted a hidden "admin" field into the form before submitting it.



```
Selenium.prototype.doAddHiddenField = function(name, value) {
  var form = this.page().getDocument().getElementsByTagName("form")[0]
  form.innerHTML += "<input type='hidden' name='" + name + "' value='" + value + "' />";
}
```

open	/login/add_user	
addHiddenField	user[admin]	true
type	user_firstname	Dr

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

- Session fixation
 - ✓ One person (attacker) sets (and exploits) another user's session id
- Test:
 - ✓ If I set the id of the session cookie, will the application accept it?

Select session id(misuse)		
open	/selenium/setup?fixtures=all	
open	/	
createCookie	_OOPSLA2007_session_id=445801d7597b50a14614a4d7d04c275c	path=/ path=/
open	/	
assertNotCookie	_OOPSLA2007_session_id=445801d7597b50a14614a4d7d04c275c	

■ Test:

- ✓ When I log in (my privileges are elevated), will my session id change?

Keep session id when auth(misuse)		
open	/selenium/setup?fixtures=all	
open	/	
deleteCookie	_OOPSLA2007_session_id	/
open	/	
storeValue	cookie=_OOPSLA2007_session_id	oldCookie
click	link=Log in	
type	name=name	gandalfw
type	name=password	mellon
clickAndWait	name=commit	
assertNotValue	cookie=_OOPSLA2007_session_id	\${oldCookie}

- Security in an agile project
- Misuse stories – the evil counterparts of user stories
- Enter Selenium
- The demo application
- Examples
 - ✓ Testing for cross site scripting (XSS)
 - ✓ Testing for cross site request forgery (CSRF)
 - ✓ Testing for insecure input handling
 - ✓ Testing for session fixation
 - ✓ Testing for information leakage

- Scan page source for words and phrases
 - ✓ "debug", "filename", "password", "SQL"
- Wrote a new Selenium assertion "assertNotExists"
 - ✓ Word or phrase does not exist in the page source code (including comments)

View debug info(misuse)	
open	/selenium/setup?fixtures=all
open	/
assertNotExists	debug

Testing for information leakage (2)

- Partial tests – reuse test code

```
_look_for_leakage.sel x  
|assertNotExists|debug|  
|assertNotExists|filename|  
|assertNotExists|sql|  
|assertNotExists|trace|
```

```
navigate_blogs.sel x  
|open|/selenium/setup?fixtures=all|  
|open|/|  
|includePartial|partials/look_for_leakage|  
|clickAndWait|link=The life of a Wizard|  
|assertTextPresent|Posted by Gandalf|  
|includePartial|partials/look_for_leakage|  
|goBack|
```

clickAndWait	link=The life of a Wizard	
assertTextPresent	Posted by Gandalf	
assertNotExists	debug	
assertNotExists	filename	
assertNotExists	sql	
assertNotExists	trace	
goBack		
assertTitle	glob:OOPSLA*	
clickAndWait	link=Boring Springfield	
assertTextPresent	Posted by Bart	
assertNotExists	debug	
assertNotExists	filename	
assertNotExists	sql	
assertNotExists	trace	

- We have created rather simple tests that manifest vulnerabilities
- We have leveraged a general testing tool
- We have discovered, tested, and fixed security issues incrementally

Questions?

BEKK

Contact me:

vidar.kongsli [at] bekk.no

■ Test:

- ✓ Will SSL be enforced on pages where sensitive information is transferred?

<i>Access login without ssl(misuse)</i>		
open	/selenium/setup?fixtures=all	
open	http://localhost/login/login	
assertTextPresent	Please log in	
assertSsl		

```
Selenium.prototype.assertSsl = function() {  
    var doc = this.page().getDocument();  
    Assert.matches("https:", doc.URL.substr(0, 6));  
};
```

- Selenium (core) is JavaScript-based
 - ✓ Sandbox model
 - ✓ An insecure page (non-SSL) cannot access a secure page (SSL)
 - ✓ A page can only interact with pages from the same origin as it self
- Alternative
 - ✓ Use Selenium Remote Control (RC)