AUGMENTED REALITY IN YOUR WEB PROXY

Roberto Suggi Liverani - @malerisch
ISC(2) / OWASP Belgium Chapter
Leuven – 17 December 2013

Who am I?

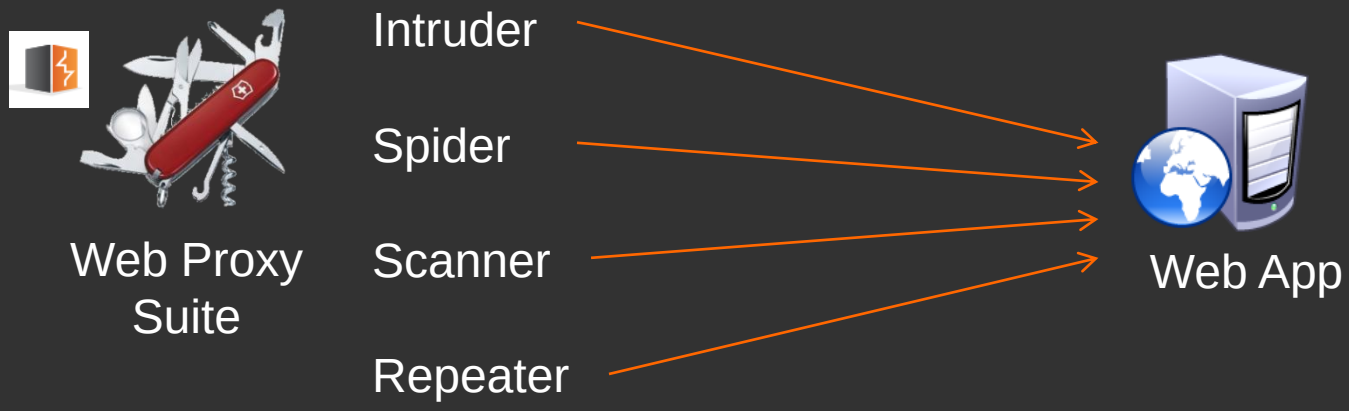
- A guy who likes to find bugs ☺
- Speaker at various cons/events:
  - Hack in the Box, DefCON, EUSecWest, OWASP, HackPra AllStars
- OWASP New Zealand Founder
- Twitter: @malerisch
- Research blog: blog.malerisch.net

Outline

- Challenges / Solutions
- Introducing Burp CSJ / DEMOs
- Stories
- Conclusions / Future plans

Traditional testing approach

The concept of proxy suite



The problem is...

Web proxy originally design to focus on server-side technology

Web App

Web Proxy

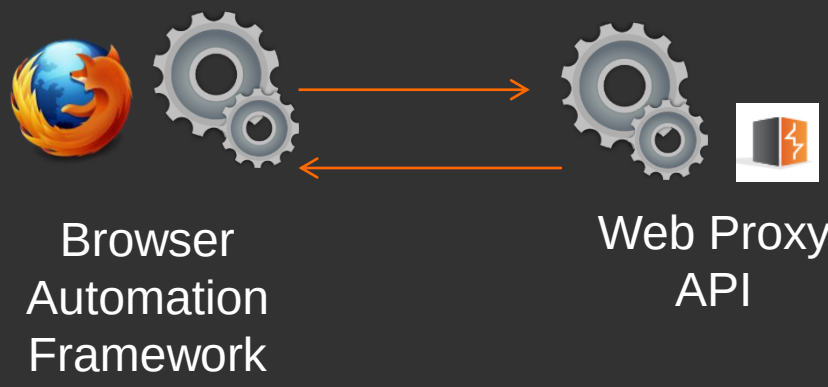
Web App

Browser

Client-side technology shift
A web app is designed to be used by a browser

Combining technologies

- *How can we get a browser close to a web proxy or vice versa?*

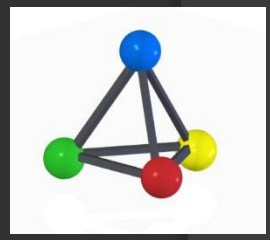


So what do we achieve?

Browser automation options...

- Selenium
  - Browser automation framework

- Crawljax
  - Crawler for Ajax apps based on Selenium



- JUnit
  - Testing framework

Selenium Server

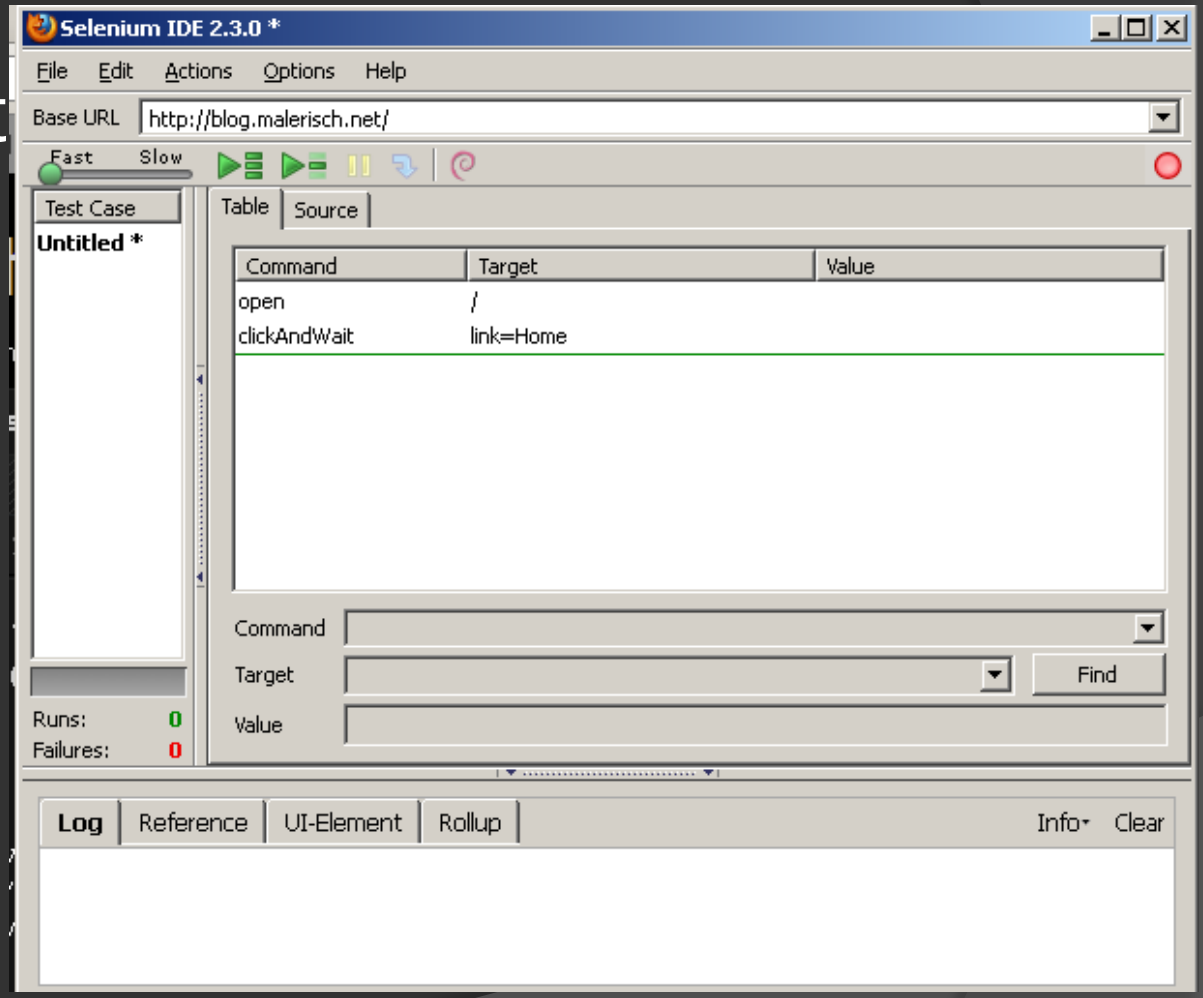
- Integrates Selenium RC
- Launches and kills browsers
- Interprets and runs Selenese commands
- Supports Grid and nodes
- Known as:
  - selenium-server-standalone
  - selenium-server

Selenium Client & WebDriver

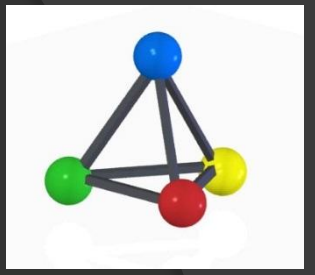
- Based on WebDriver wire protocol – RESTful + JSON
- Direct calls to browser
- Multiple drivers available: Chrome, IE, Opera, Android, iPhone
- Known as selenium-java

Selenium IDE

- Create/Repeat Execute Test case
- Firefox addon
- Export to various formats Junit WebDriver



Crawljax



- Based on Selenium WebDriver APIs
- State-flow interpretation of DOM states

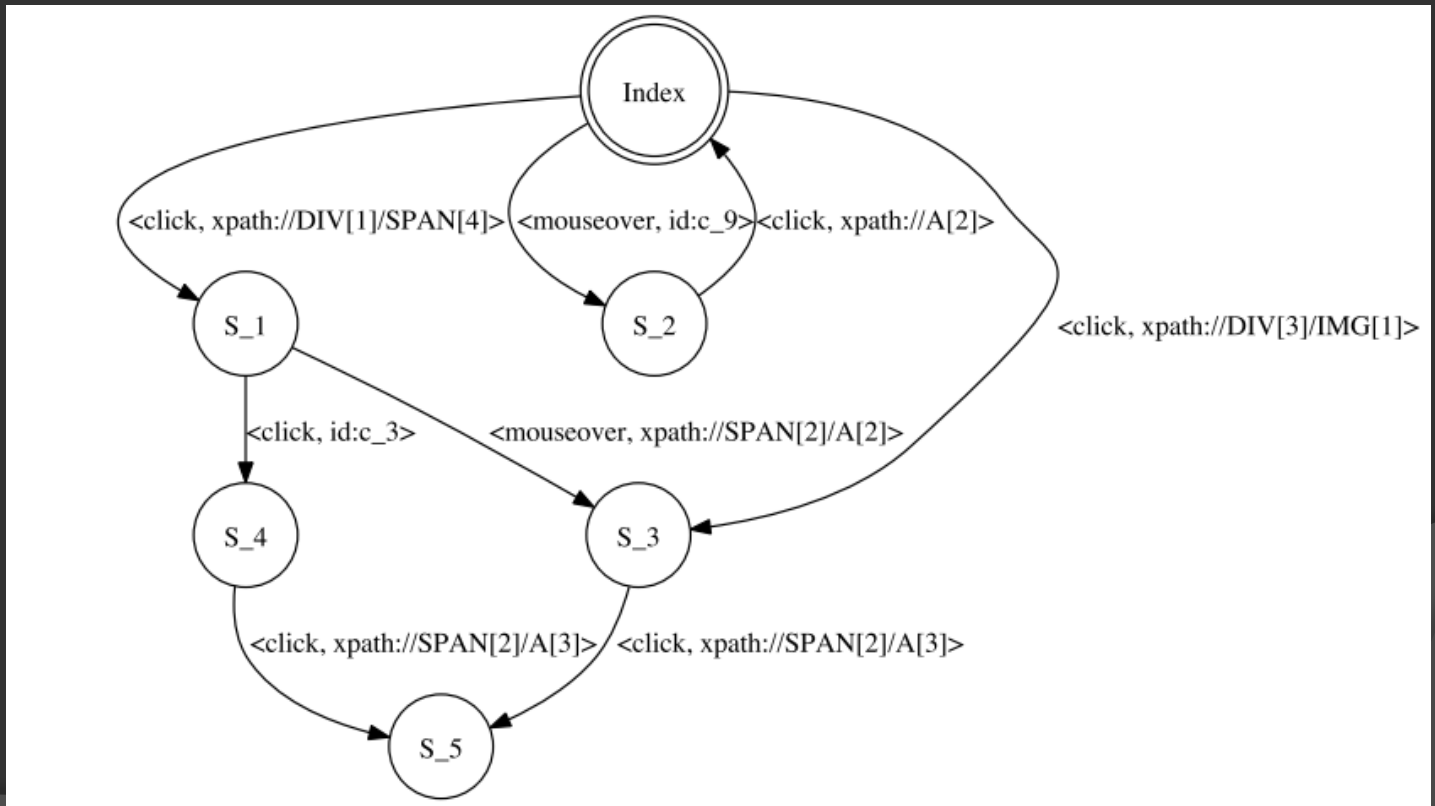


Fig. 2. The state-flow graph visualization.

Crawljax

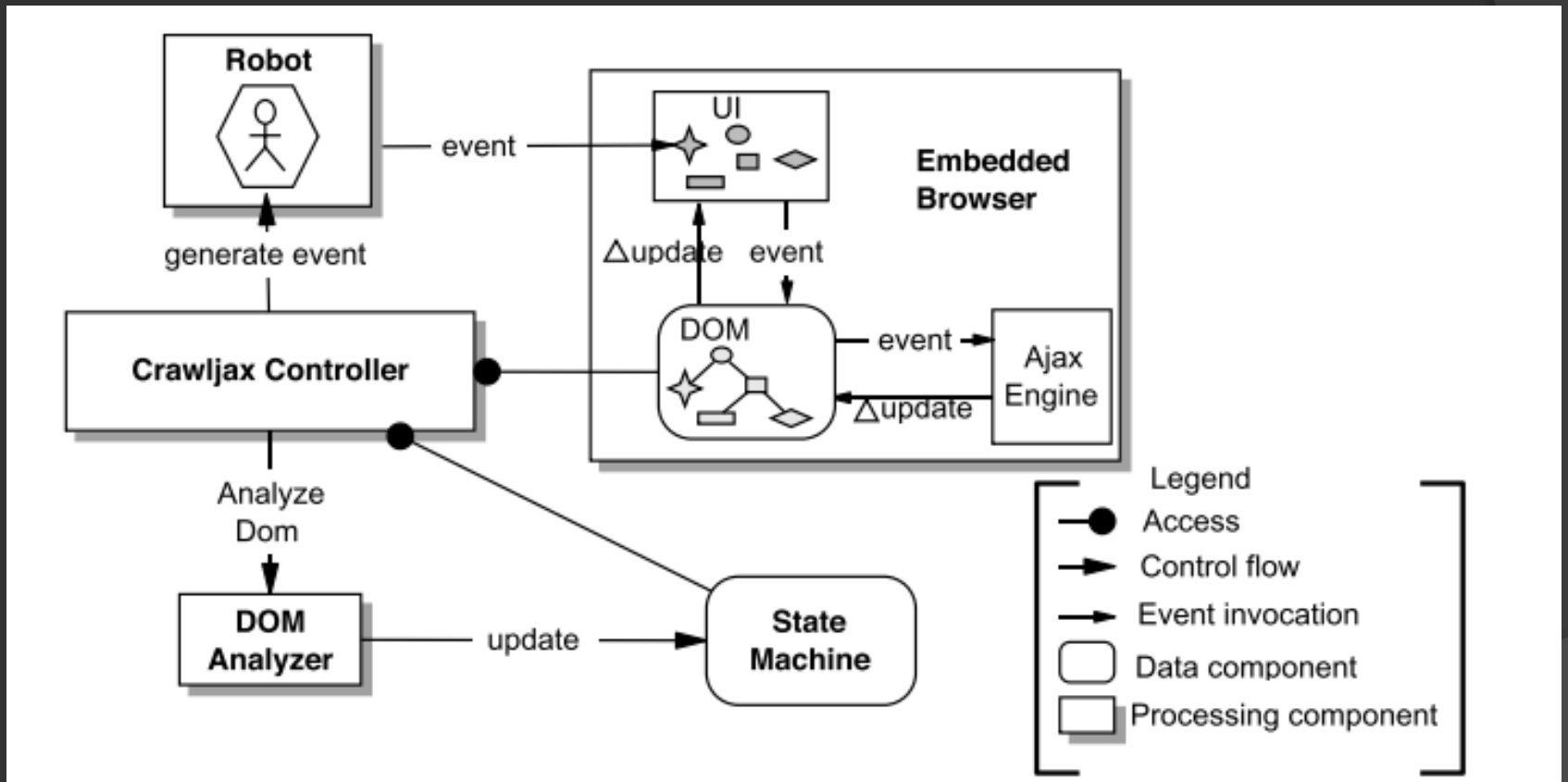
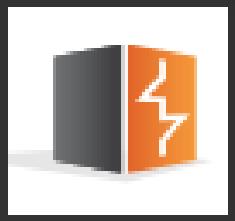


Fig. 3. Processing view of the crawling architecture.

Paper: *Crawling AJAX-Based Web Applications through Dynamic Analysis of User Interface State Changes*

Web proxy options...



- Burp Extender API
  - Java/Python/Ruby
  - Scanner, Proxy, Repeater, Cookie, Target Session handling, HTTP requests/responses

- ZAP API
  - REST interface
  - Spider, core, params, ascan, context auth, acsrf, autoupdate, pscan

Crawljax - Pros

Why integrate Crawljax?

- Augmented reality in your proxy
- Increased coverage for complex web apps
- Scalability with big/dynamic apps
- Integrated in ZAP - Ajax Spider
  @GuifreRuiz - very cool work! ☺

JUnit - Pros

Why use JUnit?

- Increase chances to discover hard-to-find bugs
- Easily create repeatable sequence of steps
- Reuse existing JUnit test-case
- Leverage Burp macro capability

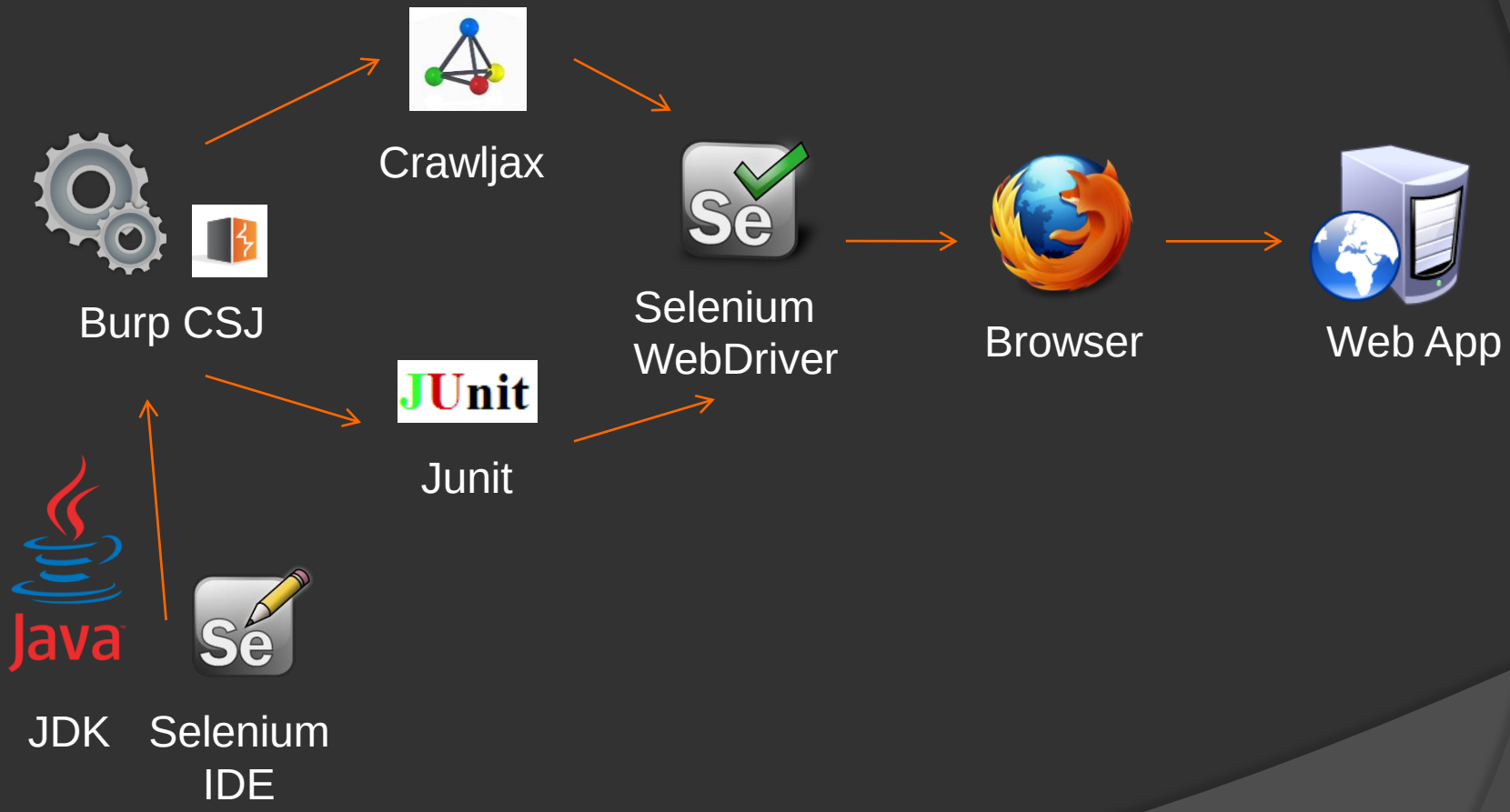
So how to combine all this?

- Created a burp extension (Burp CSJ)
  - Integrates Crawljax
  - Integrates JUnit test-case created via Selenium IDE

Source: https://github.com/malerisch/burp-csj

Coded in Java using google, stackoverflow, a mix of guessing , luck and a lot of swearing…

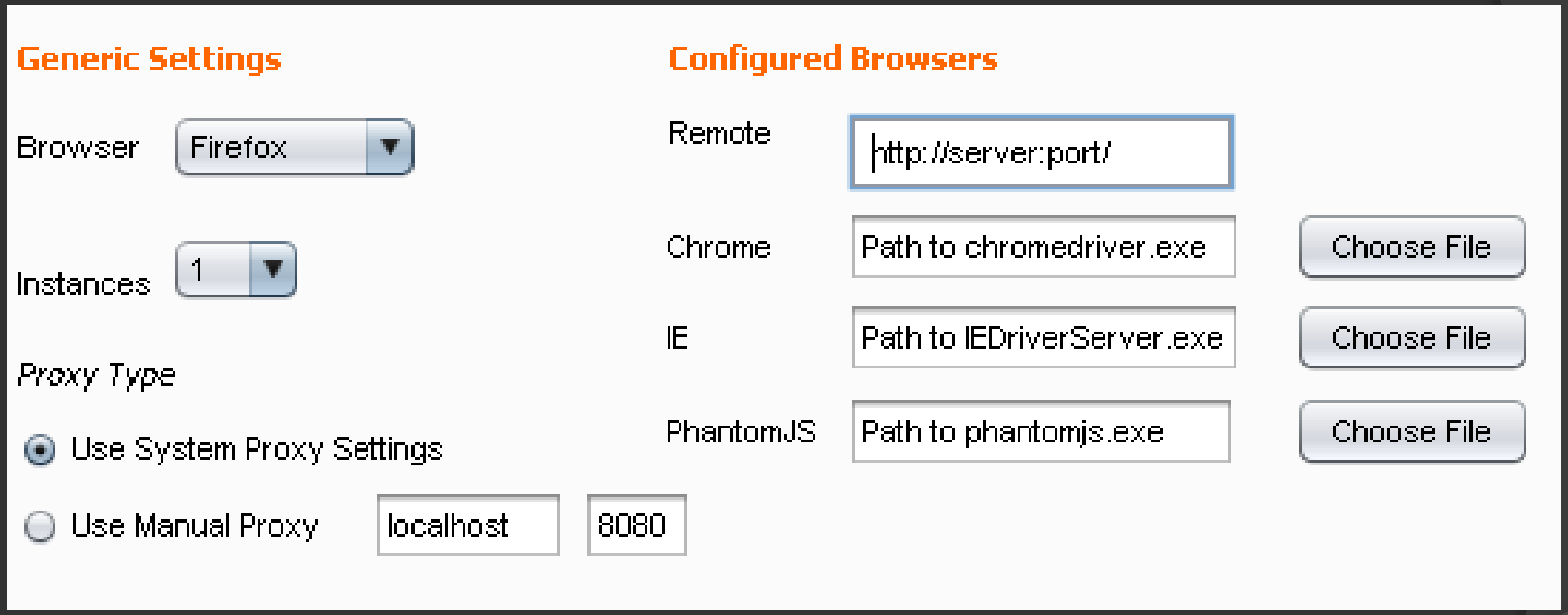
How it works...


Crawljax
Burp CSJ
Selenium
WebDriver
Browser
Web App
JUnit
Junit
Java
JDK
Selenium
IDE

Crawljax integration

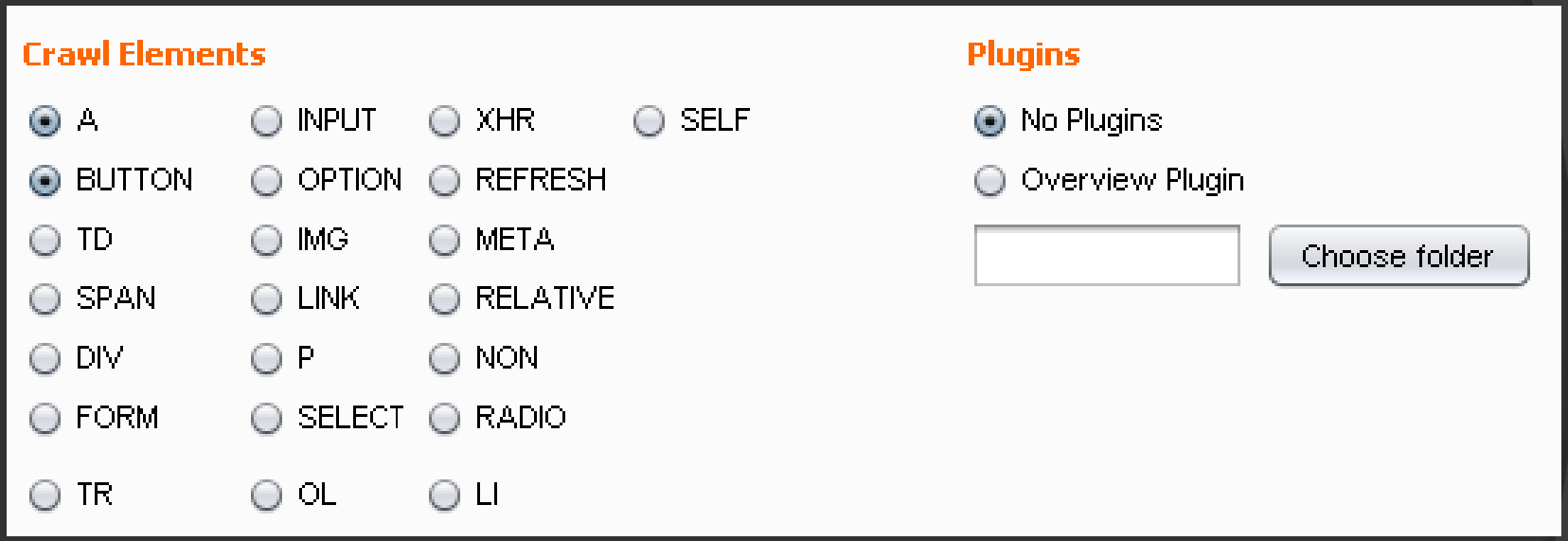
- Key Features
  - Support for Burp cookie jar
  - Support for multiple browsers, including remote webdriver
  - Support for multiple HTML elements
  - Exclusion list for crawling
  - Support for CrawlOverview plugin

Crawljax Tab (1/3)



Crawljax Tab (2/3)

Crawljax Tab (3/3)



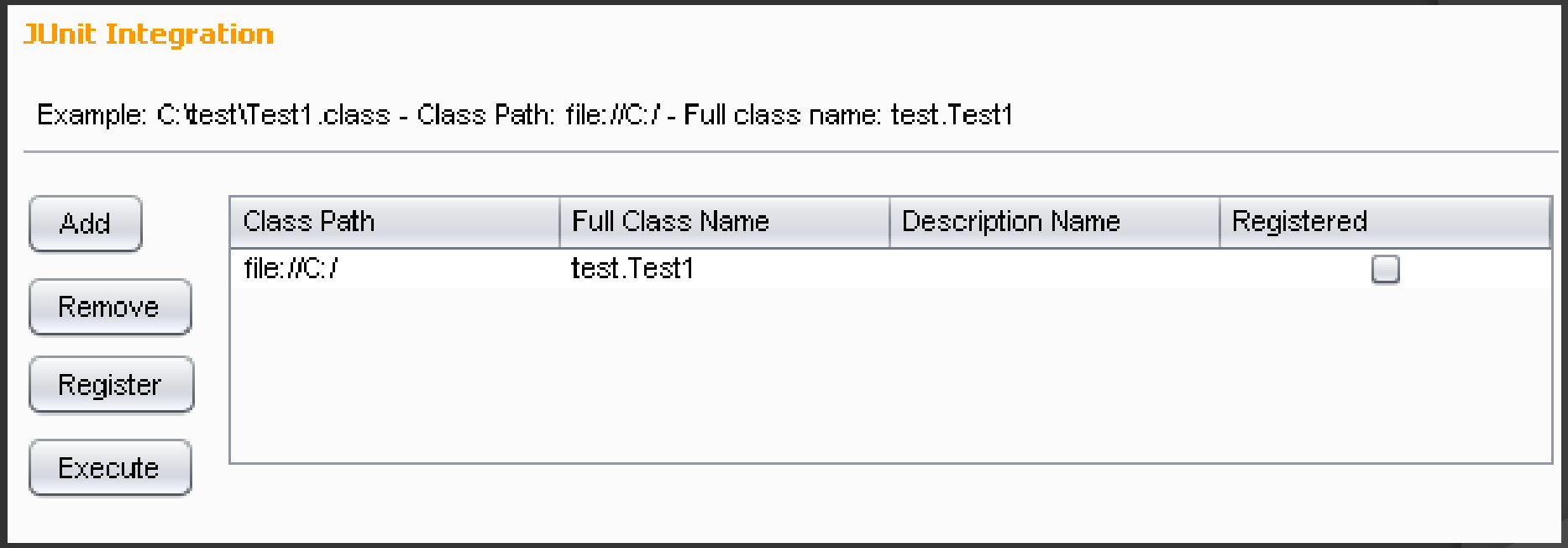
DEMO

- Crawling a site with auth
- Crawling a site with auth + remote web driver
- DEMOs
- https://www.youtube.com/watch?v=x51jwZ1HV9E
- https://www.youtube.com/watch?v=X3mjheLJEFE

JUnit Integration

- Key Features
  - Import compiled Selenium IDE JUnit Test cases
  - Register test-case into Burp session handling
  - Test case can be invoked in the Macro editor
  - Interface to execute Junit test case

JUnit Tab


JUnit Integration
Example: C:\test\Test1.class - Class Path: file://C:/ - Full class name: test.Test1
Add
Remove
Register
Execute
Class Path
Full Class Name
Description Name
Registered
file://C:/
test.Test1

DEMO

- Launching JUnit test-case via Burp Proxy
- Registering Junit Test-case via Burp and setting a macro
- DEMO: https://www.youtube.com/watch?v=mKOD3ysiN-U

Burp CSJ Tips

- Use Burp Spider + Crawljax for crawling and after scanning/attacking application
- Create JUnit test cases for sequence which takes long time to repeat
- Set Burp macro to use precisely JUnit test case
- When using Junit with Burp CSJ, set the Cookie: header with Burp

Stories from the automation world…

base64 and command injection

- Crawljax clicked on some pages with base64 data
- A scan was run before
- Some of those pages content was decoded
- Trace of ping command output were found
- An indirect OS command injection was found!

jQuery, toggle() and XSS ☺

- Complex app – use of jQuery
  - Lot of clickable elements which would invoke toggle()
- Crawljax clicked element
- New page added to Burp Target
- Page vulnerable to XSS

A nice shopping cart!

- Vulnerable shopping cart
  - Special product item would decrease amount
- Sequence of steps had to be performed before
- JUnit test-cases made the difference

Conclusions

- Combining automation is a different type of testing
  - Time for preparation needed
  - Not ideal for testers looking for quick wins
- ROI is always in bugs discovery
  - … especially bugs with critical severity

Burp CSJ future

- Resolve github tickets!
- Expand Crawljax integration
  - Support plugin import feature
- Expand JUnit Integration
  - Compile from Java Source directly…
  - Also change browser set in Junit test case…
  - Support for Burp cookie jar

Questions?

Roberto Suggi Liverani - @malerisch
blog.malerisch.net

- Source Code: https://github.com/malerisch/burp-csj
- Tutorial: http://blog.malerisch.net/2013/09/burpcsj-tutorial-using-crawljax.html

References

- Blog – Roberto Suggi Liverani
  - http://blog.malerisch.net/
- Twitter account - @malerisch
  - https://twitter.com/malerisch
- Crawling AJAX-Based Web Applications through Dynamic Analysis of User Interface State Changes
- http://www.ece.ubc.ca/~amesbah/docs/tweb-final.pdf

References

- Crawljax
  - http://crawljax.com/
- Selenium
  - http://docs.seleniumhq.org/
- JUnit
  - http://junit.org/

References

- Burp Extender API
  - http://portswigger.net/burp/extender/api/index.html
- ZAP API
  - https://code.google.com/p/zaproxy/wiki/ApiDetails
- Ajax spider in ZAP
  - https://code.google.com/p/zaproxy/wiki/GSoC2012_PluginACT