Smartfuzzing The Web

Carpe Vestra Foramina

Friday, August 5, 2011

Us

Nathan Hamiel

Principal Consultant at FishNet
Associate Professor of Software Engineering at UAT

Gregory Fleischer

Senior Consultant at FishNet

Seth Law

Principal Consultant at FishNet

Justin Engler

Consultant at FishNet

Overview

- Problems with current tools
- Current workarounds
- Proposed solutions
- Introduction to RAFT

We Aren't…

- Going to beat up on a particular vendor
  - Although we may be tempted ;)
- Solve every problem outlined in this talk
  - Although we are working on them
- Sell you a solution
  - These are people / technology problems

Our Goals

- Raise awareness
- Put focus back on the individual tester
- Get you to submit bug reports and feature requests for RAFT

Clarification

VS

Testing Tools Are Lacking

- Hey, Y2K Just called
  - Semi-automated tools fall down
  - Session and state problems
  - Problems with complicated applications
- What about modern technologies?
  - CSRF tokens and randomized DOM
  - RIA, AJAX, and Web Services

The Problems Continue

- Import of externally collected data
- Typically no analysis of results
  - Current request
  - Previous requests
  - HTTP is stateless, but analysis shouldn't be
- Testers need interaction not abstraction

The Problems Continue

- Missing “hidden” portions of the application
  - “Accept” Header manipulation

```
GET /viaf/75785466/ HTTP/1.1
Host: viaf.org
Accept: application/rdf+xml
```

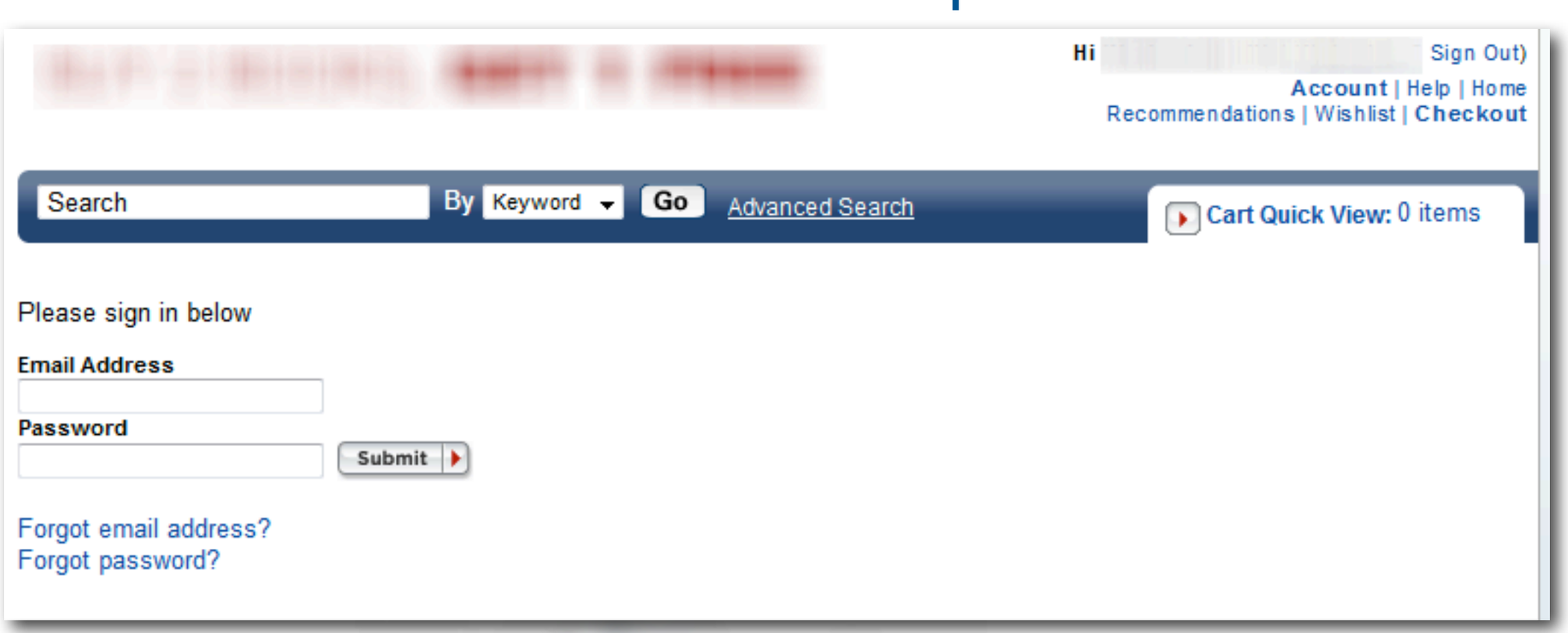
!=

```
GET /viaf/75785466/ HTTP/1.1
Host: viaf.org
```

  - User-Agent

And again

- Difficult cases
  - Risk based logins
  - In-session detection
  - Confirmation on next step



It's The Simple Things

- Missing simple features
  - Request time
  - Authorization checks
  - Client-side storage locations

Mo Tools, Mo Problems

- External tools and custom scripts
  - Can be painful with no analysis help
  - Request/response diffs
  - Full request/response logging?
- Data in multiple sources
  - No cross-tool analysis
  - Limited ability to find “new” bugs in old data

Current Solutions

- Test manually
  - Totally not time consuming, at all!
  - Modify existing tools for purposes which they weren't intended
    - Custom one-off tools and scripts
- End up missing the point
  - Results in custom formats
  - Common vulns can be missed

So Adapt Or...

- Some tools need to adapt or become useless

Want Some Examples?

- Still using Nikto?
  - What about a tool based on Nikto?
  - Do you know what your Nikto is doing?
- What about your Buster?
  - DirBuster!

A Word on Word Lists

- The DirBuster word lists have not been updated since 2007
- Many common directories and words missing from the lists
- Generating a list from spidering websites leads to poor word values
  - Only find the directories that site exposes
  - SEO poisoning can skew results

What Have You Been Missing?

- Not in DirBuster small and medium list:
  - aspnet_client
  - _notes
  - _vti_cnf
  - _vti_log
  - _vti_pvt
  - App_Code
  - App_Data
  - pkginfo
  - _vti_bin

DirBuster WTF?

- When was the last time you found Jeremiah Grossman on your web site?

```
$ grep -i -n 'j.*grossman' directory-list-2.3-medium.txt
218265:jeremiahgrossman
218267:jgrossman
218269:http%3A%2F%2Fjeremiahgrossman
$
```

RAFT Word Lists

- Generated from “robots.txt” files
- Word selections based on Disallow
- Requested “robots.txt” from 1.7 million sites

- Alexa and Quantcast Top Million data sets
- Almost 1 million files processed
- About 350 thousand unique files
- SEO pharma and mortgage excluded

Remember

- Tools don’t find vulnerabilities, people do

A Web Smart Fuzzer?

Web Smart Fuzzer Components

- Session Management
  - Without need for complex user interaction
  - Shared cookie jar object
  - Proper in-session detection
- Sequence building and running
  - Login sequences / clean up sequences
  - Multi-stepped operations
  - Grabbing data from previous requests

Web Smart Fuzzer Components

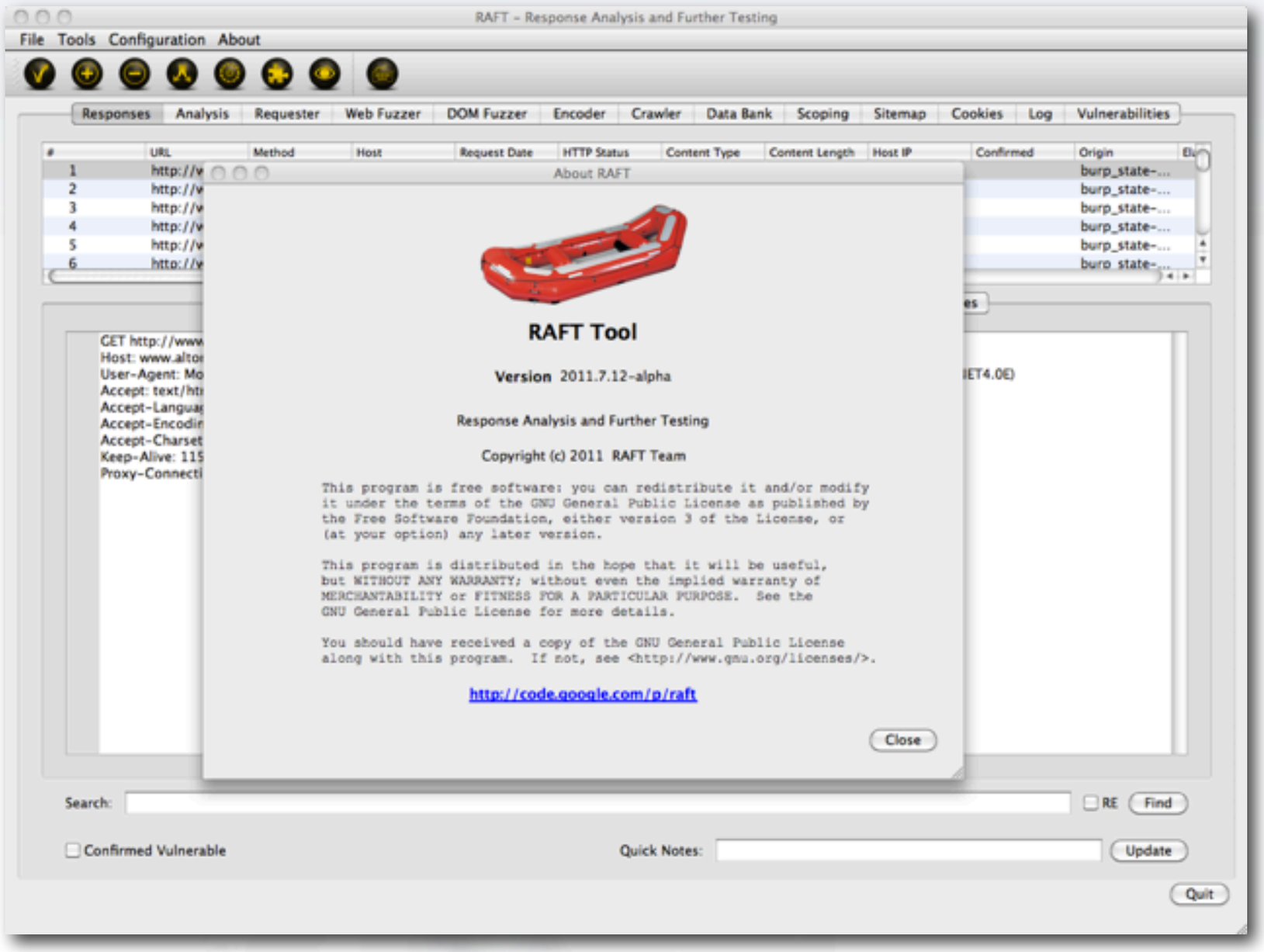
- Content discovery
  - Intelligent spidering
  - Intelligent form submission
  - Content discovery based on contextual Info
- Support modern technologies
  - HTML5
  - RIA

Web Smart Fuzzer Components

- Randomization handling
  - CSRF tokens
  - DOM data
- Payload choices
  - Based on context awareness
- Tight integration of components
- Ability to easily experiment

RAFT

- Introducing RAFT



What is RAFT?

- Response Analysis and Further Testing
- RAFT is different
  - **Not an inspection proxy!**
  - Focus on workflow
  - Analysis for other tools and scripts
- Open source (Python and QT)

Framework and Concepts

- Tool for testers
- Understands output from other tools (import from automated scanners and proxies)
  - RAFT is not an inspection proxy
- Technical Components
  - Built-In Web Browser (WebKit)
  - Fuzzers (two for the price of one!)

Platforms

- Mac OS X
  - 10.5 / 10.6
  - 10.7 probably fine with Macports
- Linux
  - Ubuntu 10.4 LTS
  - Probably just works on everything else
- Windows
  - Windows XP / Windows 7

Dependencies

- Effort is made to keep dependencies at a minimum
  - PyQT4
  - QtWebKit
  - QScintilla
  - lxml
  - pyamf
  - pydns

RAFT Download

- Check out source from project SVN
  - http://code.google.com/p/raft
- Packages for OSX and Windows coming soon
- If you find a bug, and you will, please let us know :)

Response Analysis and F**k Testing

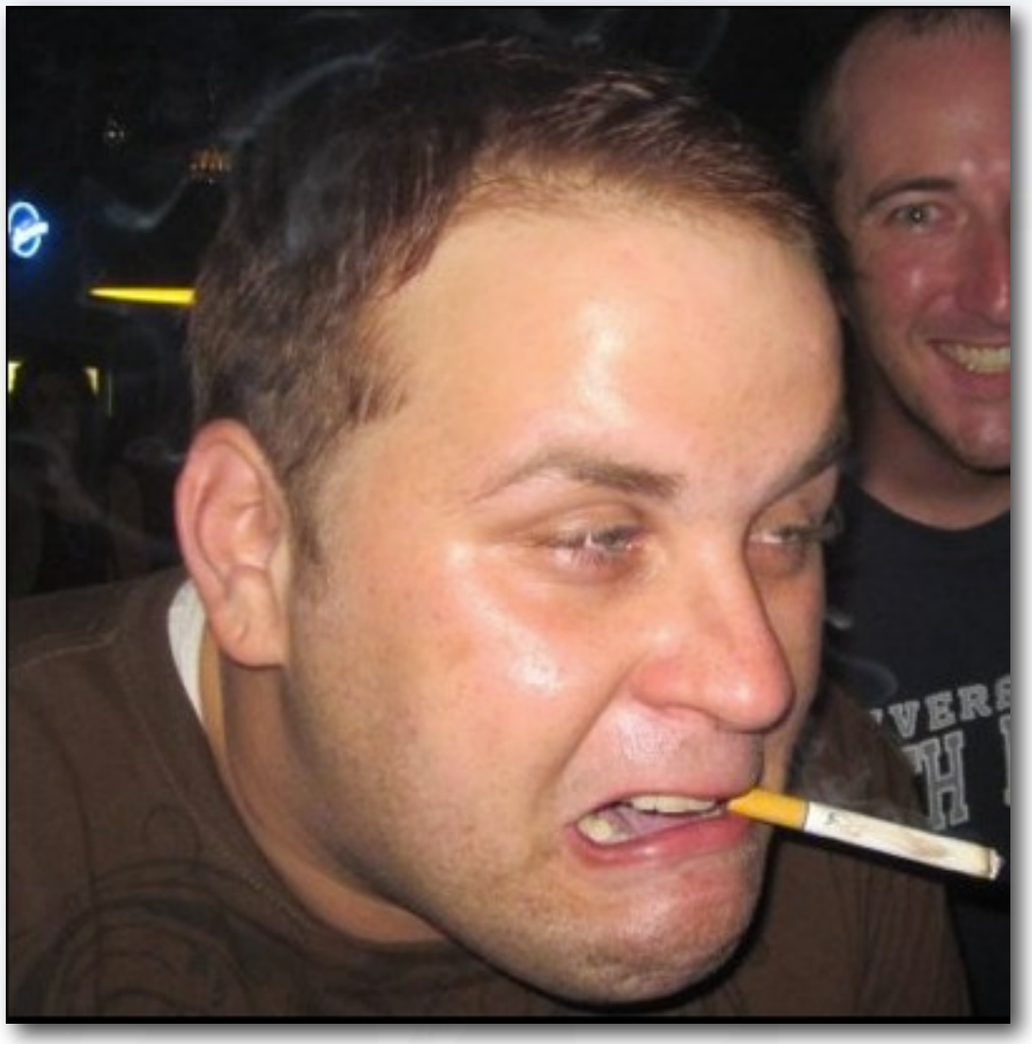
- Use RAFT for your own code
  - Modified tools and One off scripts
- RAFT Data sources
  - RAFT Capture format
  - RAFT Browser
  - Burp (Log / State / XML)
  - Webscarab
  - Paros

Interface Walkthrough

- Demo

Analysis

- Analysis - Yeah, it's kind of in the title



Analysis

- Analysis Anywhere!
  - Our concept for better tools
  - Any analysis on any data source
  - Analyzers integrated with other tools
- Modular analyzers
  - New analyzers easy to add
  - Customizable config / execution / reporting
  - Analyzers can call each other

Analysis

- Find what others ignore
  - Timing analysis
  - Same request, different response
  - Image analysis
    - Do you really want to know where your ~~facebook~~ Google+ friends have been?
  - Possibilities are endless

Analyzer Demo

- Demo

Smart Testing Components

- Templatized components
  - Requester
  - Fuzzer
- Sequence running
  - Login, cleanup, and fuzzing
- Browser object
  - For those hard to reach applications

Web Fuzzer Overview

- Templating approach
  - Allows for the tester to markup data
  - For a good time call for multiple payload types
- Pre and Post sequence running

Web Fuzzer Demo

- Demo

Sequence Fuzzer

- When you just need something more
- Allows for the insertion of data in sequences
- Dynamic data and CSRF token support

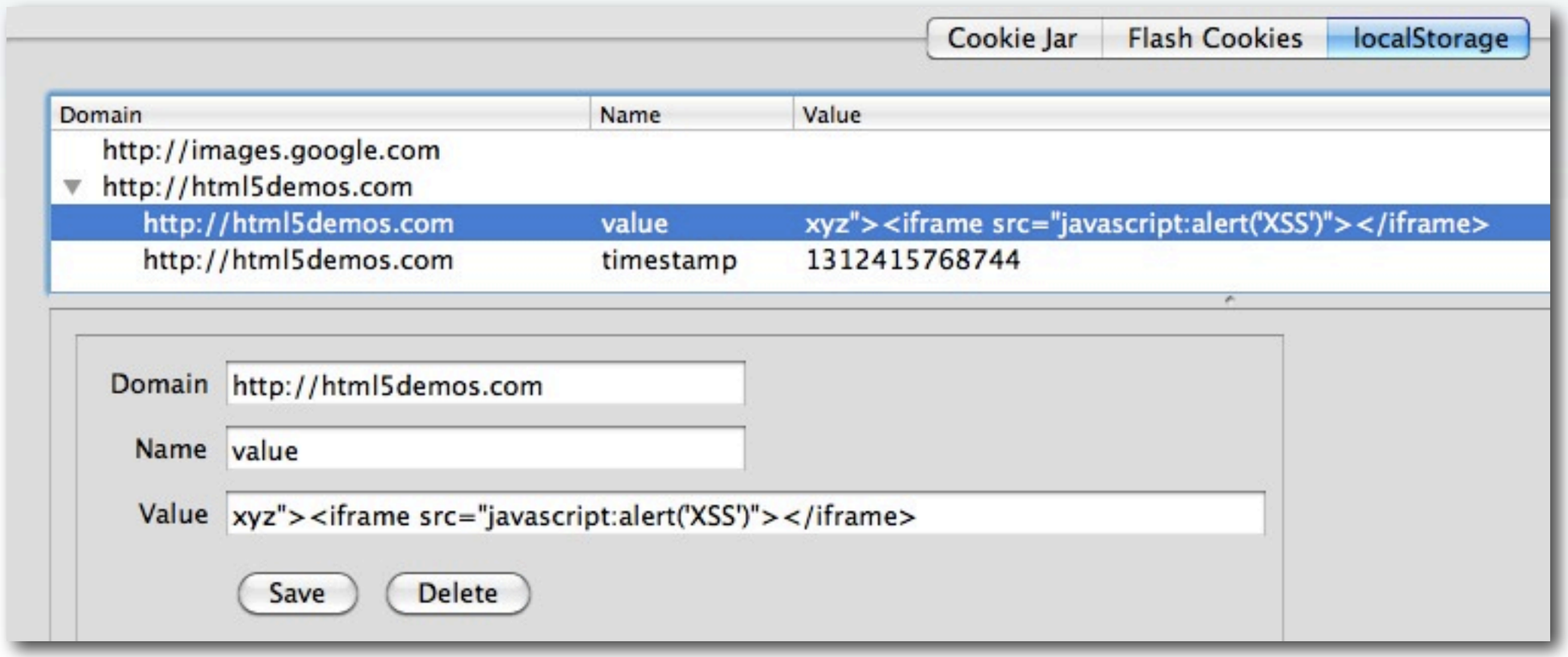
Sequence Builder

- Demo

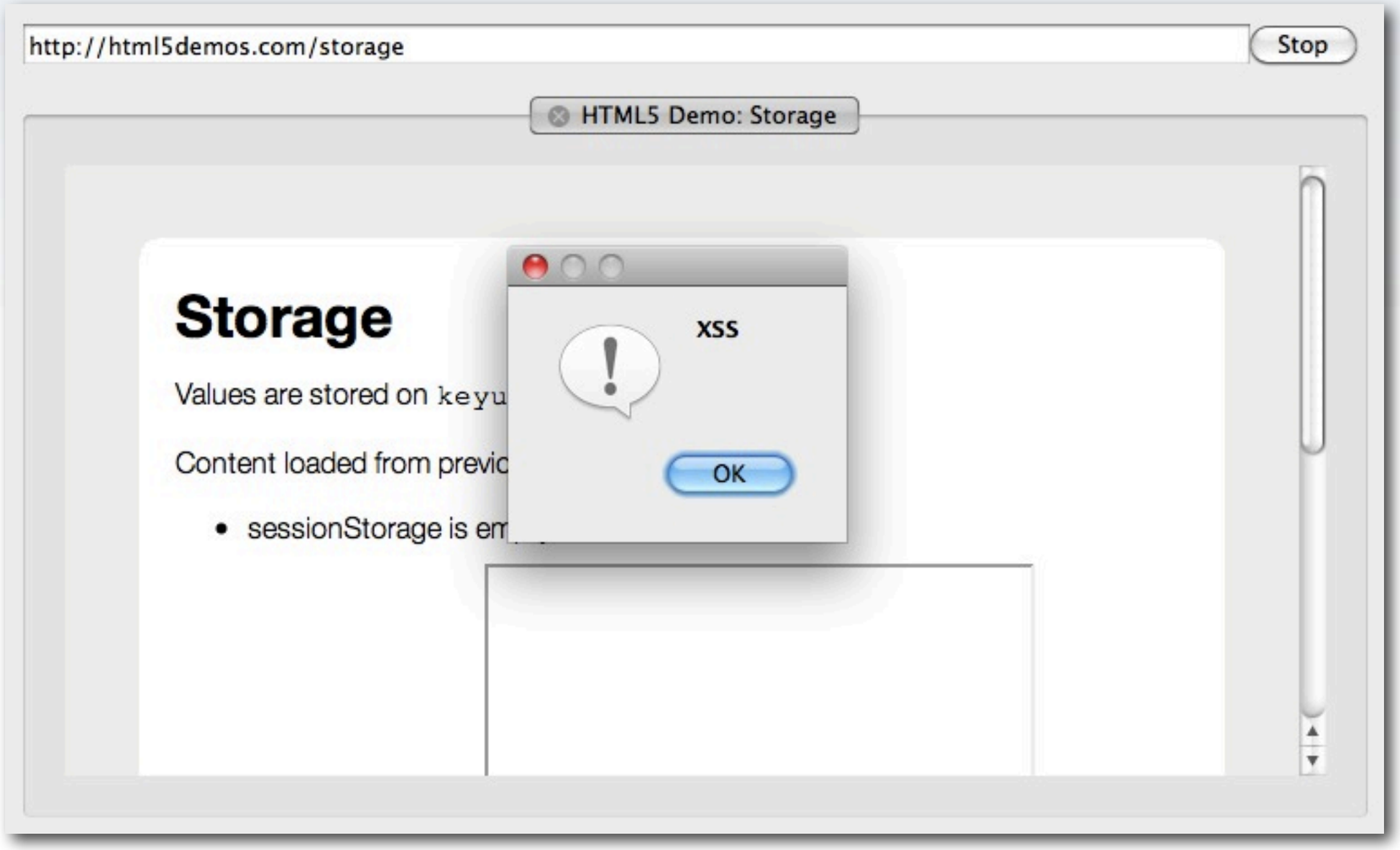
DOM Fuzzer Demo

- Demo

Deeper?


Cookie Jar
Flash Cookies
localStorage
Domain
Name
Value
http://images.google.com
http://html5demos.com
http://html5demos.com
value
xyz"><iframe src="javascript:alert('XSS')"></iframe>
http://html5demos.com
timestamp
1312415768744
Domain
http://html5demos.com
Name
value
Value
xyz"><iframe src="javascript:alert('XSS')"></iframe>
Save
Delete

Pop!


http://html5demos.com/storage
Stop
HTML5 Demo: Storage
Storage
Values are stored on keyu
Content loaded from previc
sessionStorage is em
xss
OK

Documentation

- Really?
  - Available on the wiki of the project page

Language Integration

Scala

Python

Smalltalk

Ruby

COBOL

Java

.Net

Fortran

Visual Basic

Visual FoxPro

RAFT Capture Format

```
<!ELEMENT raft (capture*)>
<!ATTLIST raft version CDATA #IMPLIED>
<!ELEMENT capture (request, response?, analysis?)>
<!ELEMENT request (method?, url?, host?, hostip?, datetime?, headers, body?)>
<!ELEMENT response (status?, content_type?, content_length?, elapsed?, headers, body?)>
<!ELEMENT analysis (notes?, confirmed?)>
<!ELEMENT method (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT host (#PCDATA)>
<!ELEMENT hostip (#PCDATA)>
<!ELEMENT datetime (#PCDATA)>
<!ELEMENT headers (#PCDATA)>
<!ATTLIST headers encoding (none|base64) "none">
<!ELEMENT body (#PCDATA)>
<!ATTLIST body encoding (none|base64) "none">
<!ELEMENT status (#PCDATA)>
<!ELEMENT content_type (#PCDATA)>
<!ELEMENT content_length (#PCDATA)>
<!ELEMENT elapsed (#PCDATA)>
<!ELEMENT notes (#PCDATA)>
<!ELEMENT confirmed (#PCDATA)>
```

RAFT Capture Example

```
<raft version="1.0"><capture>
<request>
<method>GET</method>
<url>http://www.altoromutual.com/</url>
<host>www.altoromutual.com</host>
<datetime>Thu May 12 21:16:07 2011 GMT</datetime>
<headers>GET / HTTP/1.1
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/533.3 (KHTML, like Gecko) Qt/4.7.1 Safari/533.3
Accept: application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5

</headers>
<body encoding="base64"></body>
</request>
<response>
<status>200</status>
<elapsed>703</elapsed>
<content_type>text/html; charset=utf-8</content_type>
<content_length>9645</content_length>
<headers>HTTP/1.1 200 OK
Date: Thu, 12 May 2011 22:42:57 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Set-Cookie: ASP.NET_SessionId=wcmd5ji5cjzkwyuyuy1jcf55; path=/; HttpOnly
amSessionId=174257217159; path=/
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=utf-8
Content-Length: 9645

</headers>
<body encoding="base64">DQoNCjwhRE9DVFlQRSBodG1sIFBVQkxJQyAiLS8vVzNDLy9EVEQgWEhUTUwgMS4wIFRyYW5zaXRp
b25hbC8vRU4iICJodHRwOi8vd3d3LnczLm9yZy9UUi94aHRtbDEvRFREL3hodG1sMS10cmFuc2l0
```

Writing an Analyzer

- Example

RAFT Weaknesses

- Webkit dependence
- SQLite
- Random acts of buggery
  - Has to run in project directory
  - Hardcoded payloads
  - Absolutely no error handling

RAFT Future Features

- More analysis components
- Integrated scanner functionality
- Reporting output
- Command line interface

What We Aren't Working on

- An Inspection Proxy

Call to Action

- We need help
  - Contribute with code
  - Test and report bugs
  - Provide integration with other tools
- Future features
  - Request new features
  - Code new features yourself

???

- Questions?

Contact

RAFT Dev Team

http://twitter.com/RAFTDevTeam

Nathan Hamiel

http://twitter.com/nathanhamiel
nhamiel@gmail.com

Gregory Fleisher

gfleischer@gmail.com
twitter.com/%00<script>alert(0xLOL)

Justin Engler

http://twitter.com/justinengler

Seth Law

http://twitter.com/sethlaw
seth.w.law@gmail.com

Feedback Forms

- Please Remember to Complete Your Feedback Form