

# Smartfuzzing the Web



Carpe Vestra Foramina

# Out of Date

- This presentation is out of date.
  - Grab an updated copy from our Google Code Page
- <http://code.google.com/p/raft>

# Who are we?

- Nathan "Nate Dawg" Hamiel
- Gregory "G-Fresh" Fleischer
- Seth "The Law" Law
- Justin "J-Roc" Engler



# Presentation Overview

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

# Testing Tools Are Lacking

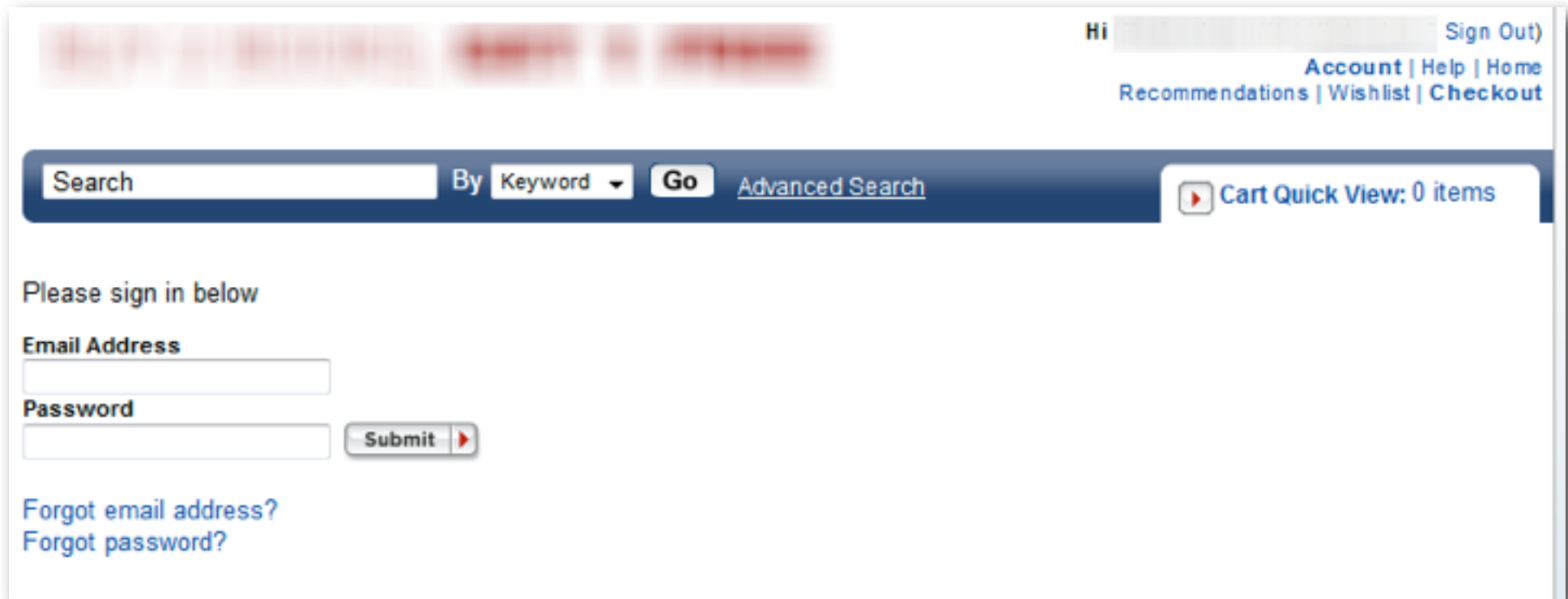
- Semi-automated web testing tools can be pretty dumb
  - Pick your poison, each one falls down at some point
  - Session data stays stale
  - State maintenance sucks or non-existent
  - Can't handle things beyond simple applications
  - Don't allow the import of externally-collected data
- What about modern technologies?
  - CSRF Tokens
  - Randomized DOM variables
  - RIA apps
  - Web services
  - JS/AJAX

# The Problems Continue

- Typically no analysis is performed
  - Testing responses alone may contain a treasure trove of vulnerabilities that are not currently identified
  - Analysis only runs on current request, what about all the old data?
  - Just because HTTP is stateless, doesn't mean our analysis has to be.
  - No vulnerability or sensitive data identification
- Testers don't need abstraction
  - APIs and difficult formats
- Missing simple features
  - Request time

# And Continue Again

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



The screenshot shows a web application interface. At the top, there is a blurred header area. On the right side of the header, the text "Hi [blurred] Sign Out)" is visible, along with navigation links: "Account | Help | Home Recommendations | Wishlist | Checkout". Below the header is a search bar with the text "Search" and a dropdown menu set to "By Keyword". A "Go" button and a link to "Advanced Search" are also present. On the right side of the search bar, there is a "Cart Quick View: 0 items" button. Below the search bar, the text "Please sign in below" is displayed. Underneath, there are two input fields: "Email Address" and "Password". A "Submit" button is located to the right of the password field. At the bottom left, there are two links: "Forgot email address?" and "Forgot password?".

# And Continue Yet Again

- External tools and custom scripts
  - Can be painful
  - No analysis
  - Request/response diffs
  - Syntax highlighting (duh)?
  - Full request/response logging
- Data in multiple tools
  - No cross-tool analysis
  - Archiving problems with collected data
  - Limited ability to find "new" bugs in old data



# What Do People Do?

- Test manually
  - First of all, Yuck!
  - Modify other tools for purposes which they weren't intended
  - Write custom tools and scripts for one-off purposes
- Can end up missing quite a bit
  - Reinventing the wheel can be hard!
  - Typically one offs
  - No in-depth analysis
  - Even common vulnerabilities can slip through the cracks
  - Results are stored in custom formats in multiple files

# Adapt Or...

- Some tools have to adapt or they become useless



# A Web Smart Fuzzer?



# A Web Smart Fuzzer

- Session management
  - Without complex user interaction
  - Shared cookie jar
- Sequence building and running
  - Login sequences
  - Multi-stepped operations
  - Grabbing data from previous page for request
- Finding content to fuzz
  - Intelligent spidering and form submission
  - Content discovery based on contextual information
- Support for modern features
  - HTML5

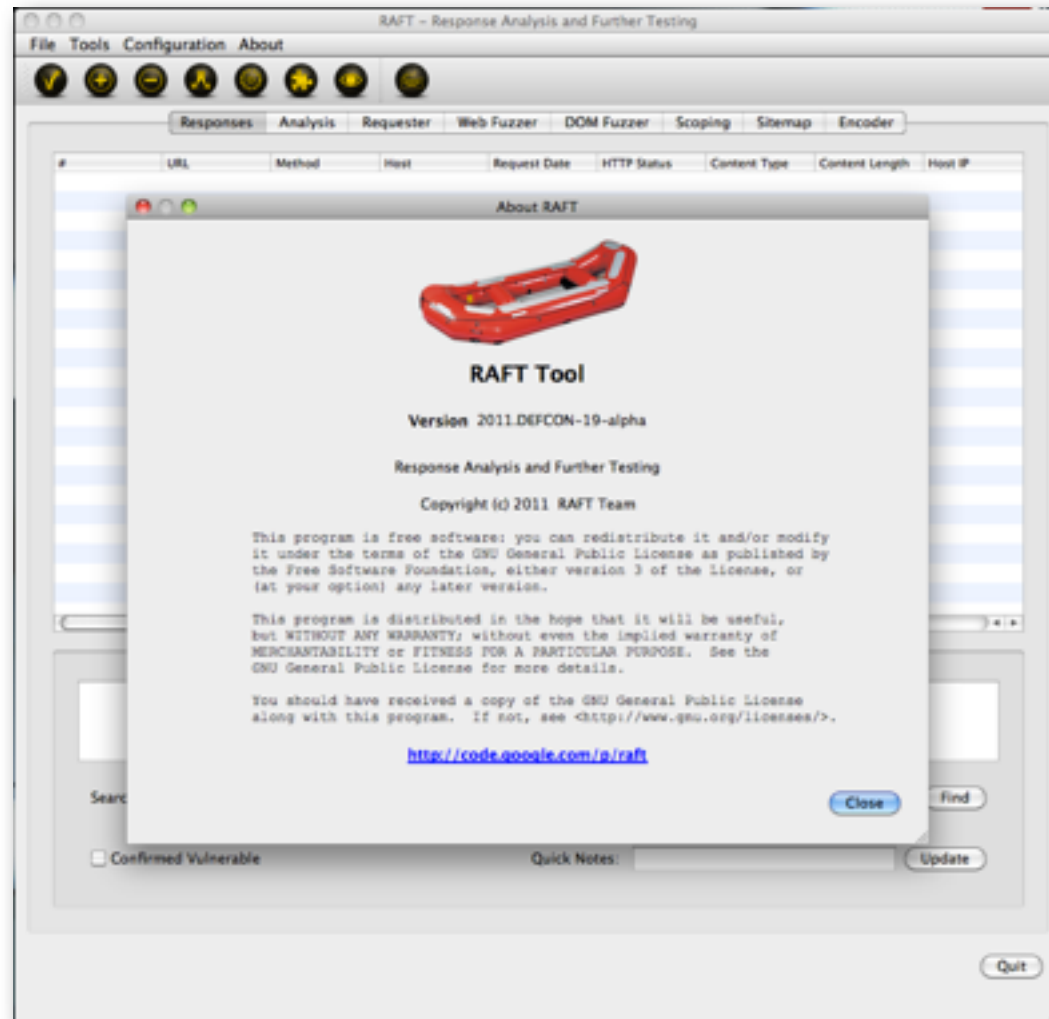
# Web Smart Fuzzer Components

- CSRF and other random data handling
  - Ability to handle CSRF tokens per page
  - Ability to handle randomized data on the DOM
- Payload choices based on context awareness

# Web Smart Fuzzer Components

- Tight integration of various components
- Ability to easily experiment with new features

# RAFT



# RAFT

- Response Analysis and Further Testing
- RAFT is different
  - Not an inspection proxy
  - Focus on workflow
  - Analysis for your own tools and scripts
  - Import data from other tools
- Open source (written in Python and QT)
- Target Platforms
  - Windows XP / Windows 7
  - Mac OS X 10.5 / 10.6
  - Linux Ubuntu 10.4 LTS



# RAFT Dependencies

- PyQt4
- QtWebKit
- QScintilla
- lxml
- pyamf
- pydns

# RAFT Download

- Check out source from SVN
- Download packages for
  - OS X
  - Windows
- <http://code.google.com/p/raft>

# Analysis

- Don't be caught without an analyzer



# Analysis

- Analysis Anywhere!
  - Our concept for better tools
  - Any analysis on any data source
  - Analyzers fully integrated with other tools in RAFT
- Modular Analyzers
  - New analyzers easy to add
  - Config, execution, and reporting all customizable
  - Analyzers can call each other
- Find the stuff that others ignore
  - Timing analysis
  - Same request, different response (no no, this /never happens)
  - Possibilities are pretty much endless

# Smart testing components

- Fuzzing, just smarter
  - Handling of CSRF tokens
  - Browser object handling
  - Sequence handling

# Documentation

- Who needs documentation... really ;)
  - Available on the wiki of the project page



# RAFT Data Formatting

- Other language integration
  - XML Capture Format
  - Python
  - Ruby
  - Perl
  - Java

# RAFT Future Features

- More Analysis
- Integrated Scanner Functionality
- Reporting Output
- Command Line Interface



# 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
- Demand better tools from commercial vendors as well

# Questions?



# Contact

**Nathan Hamiel**

<http://twitter.com/nathanhamiel>

**Justin Engler**

<http://twitter.com/justinengler>

**Gregory Fleischer**

[gfleischer@gmail.com](mailto:gfleischer@gmail.com)

[twitter.com/%00<script>alert\(0xL0L\)](http://twitter.com/%00<script>alert(0xL0L))

**Seth Law**

<http://twitter.com/sethlaw>