Smartfuzzing The Web
Carpe Vestra Foramina
• This presentation is out of date
  • Grab an updated copy from our Google Code page
  • http://code.google.com/p/raft
Us

Nathan Hamiel
Principal Consultant at FishNet
Associate Professor of Software Engineering at UAT

Seth Law
Principal Consultant at FishNet

Gregory Fleischer
Senior Consultant at FishNet

Justin Engler
Consultant at FishNet
Overview

- Problems with current tools
- Current workarounds
- Proposed solutions
- Introduction to RAFT
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
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
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
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
  - 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
Introducing RAFT
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)
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
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](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 :)  

Wednesday, July 27, 2011
Analysis

- Don’t be caught without an analyzer
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
Smart Testing Components

- Templatized components
  - Requester
  - Fuzzer
- Sequence running
  - Login, cleanup, and fuzzing
- Browser object
• 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 components
• 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
• Questions?
Contact

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

Justin Engler
http://twitter.com/justinengler

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

Seth Law
http://twitter.com/sethlaw
seth.w.law@gmail.com
Feedback Forms

• Please Remember to Complete Your Feedback Form