BURPKIT

Using WebKit to *Own* the Web

Presented by:
Nadeem Douba

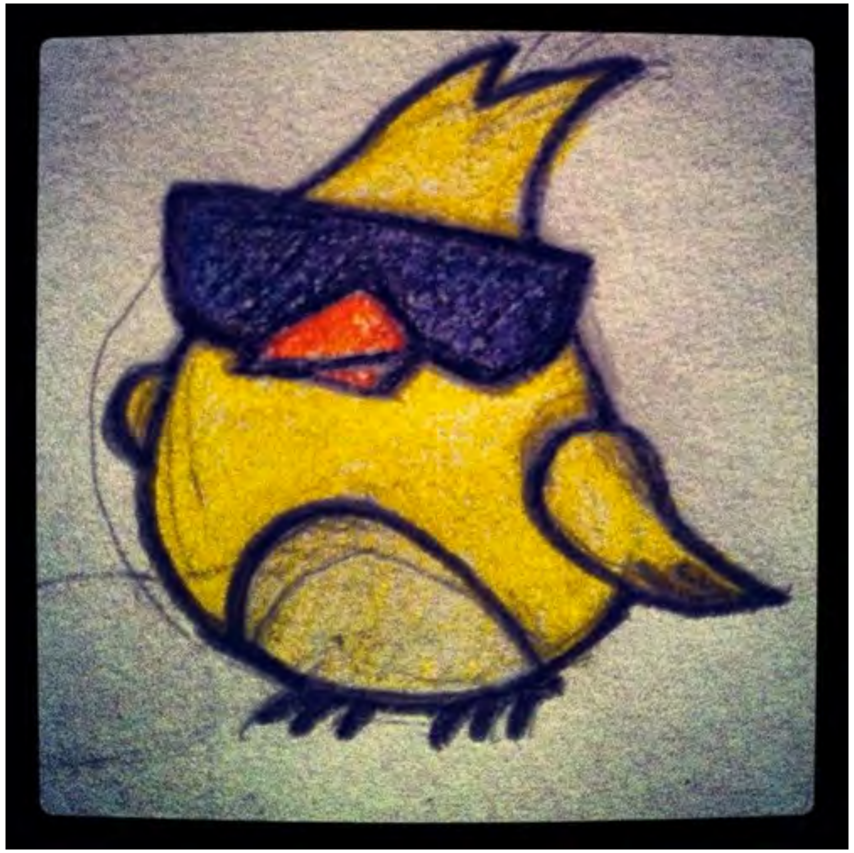
redcanari
INFORMATION SECURITY

2015-07-15

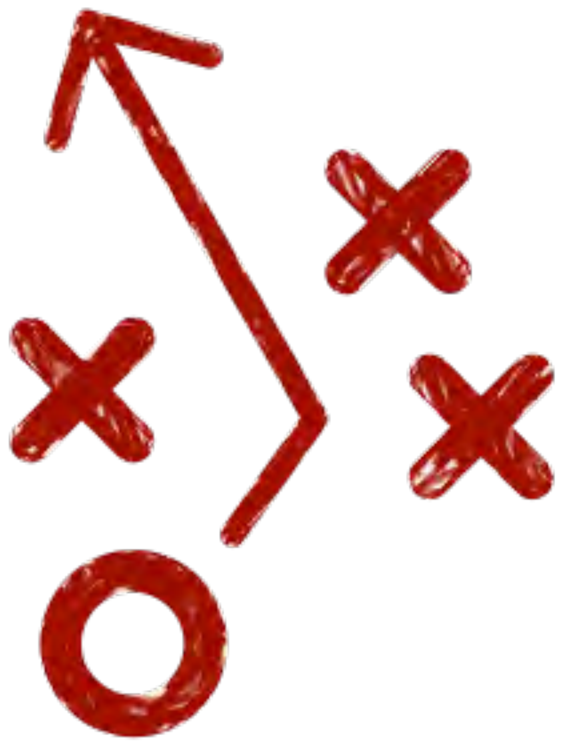
DEF CON 23 LAS VEGAS

INTRODUCTION

- **Nadeem Douba**
  - Founder of Red Canari, Inc.
  - Based out of Ottawa, ON.
  - Hacker
- **Interests:**
  - Exploiting stuff
  - Building hacking tools
- **Prior work:**
  - Sploitego (presented at DEF CON XX)
  - Canari (used by Fortune 100s)
  - PyMiProxy (used by Internet Archive)



OVERVIEW



- **WebKit**
  - What is it?
  - Why use it?
  - How can we use it?
- **BurpKit**
  - Design Considerations
  - Implementation
  - Demos!
- **Conclusion**
- **Questions?**

THE WEB PEN-TESTER'S CONUNDRUM

- Today's web applications are complex beasts
- ***Heavy*** use of JavaScript for:
  - Rendering pages
  - Rendering page elements
  - Performing web service requests
- ¿But our ***security tools*** are ***still*** scraping HTML**!?**

OUR TOOLKIT

- **Reconnaissance & Scanning:**
  - Most tools (nikto, cewl, etc.) just scrape HTML
- **Attack:**
  - **BurpSuite Pro/Community**
    - **Lobo-based** Renderer tab (Burp's neglected child) ☹
    - No JavaScript/HTML5 support
  - Charles & Zed are just proxies
  - WebSecurify's Proxy.app only has a web view

MODERN TOOLKIT REQUIREMENTS

- Web penetration testing tools that:
  - Have modern web browser capabilities
  - Parse and interpret JavaScript
  - Dynamically render and inspect content
- **Most importantly:**
  - Our tools need to be able to interact with the DOM!

WEBKIT

What is it good for? - Lots of things!

WHAT IS WEBKIT?

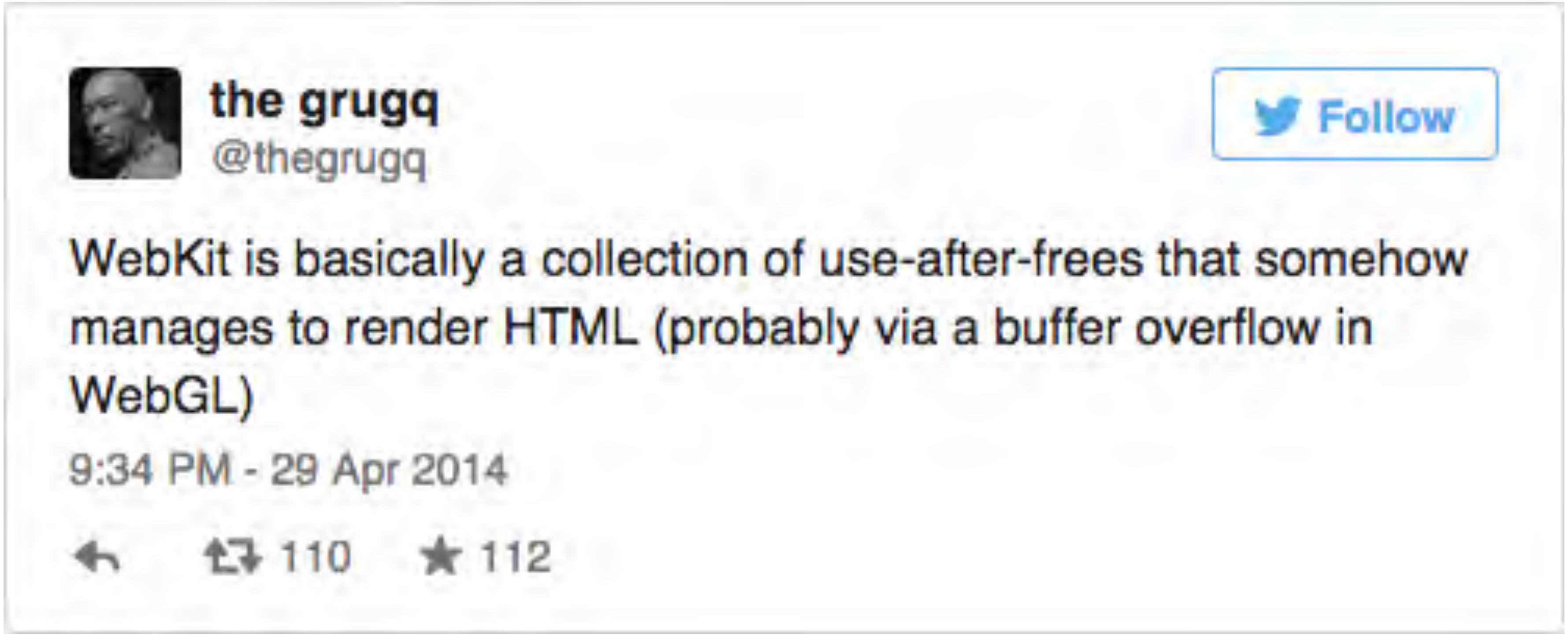
"**WebKit** is a layout engine software component for rendering web pages in web browsers. It powers Apple's Safari web browser, and a fork of the project is used by Google's Chrome web browser."

- Wikipedia (https://en.wikipedia.org/wiki/WebKit)

Image credit: Smashing Magazine

redcanari INFORMATION SECURITY

(UN)OFFICIAL DEFINITION…



the grugq
@thegrugq

Follow

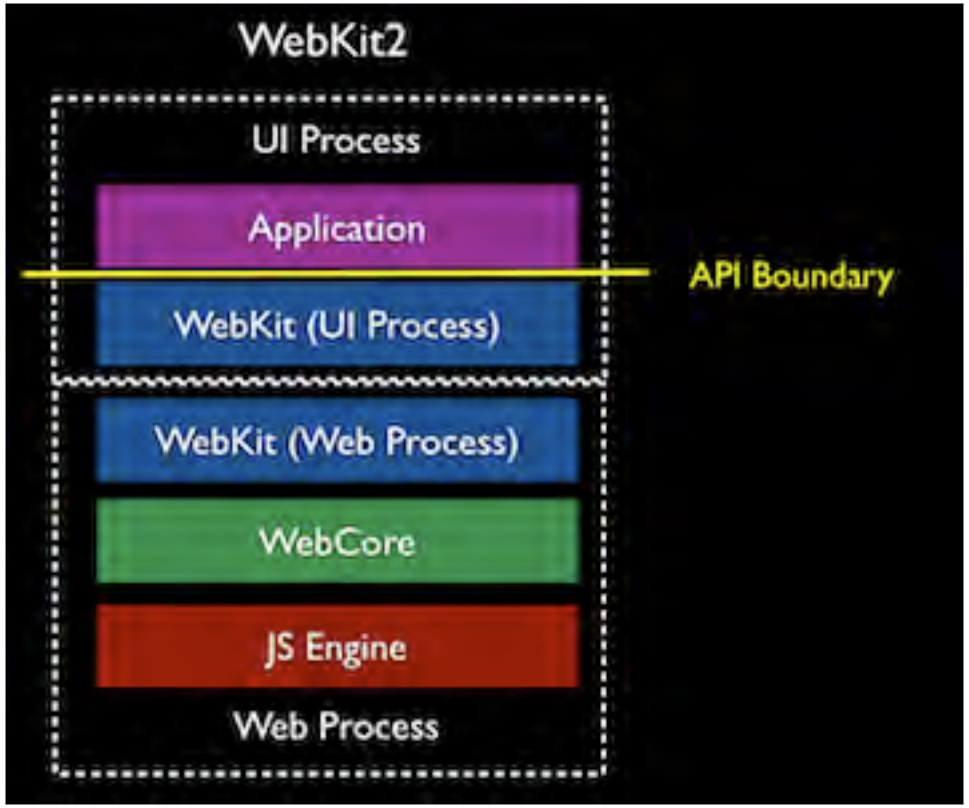
WebKit is basically a collection of use-after-frees that somehow manages to render HTML (probably via a buffer overflow in WebGL)

9:34 PM - 29 Apr 2014

110 112

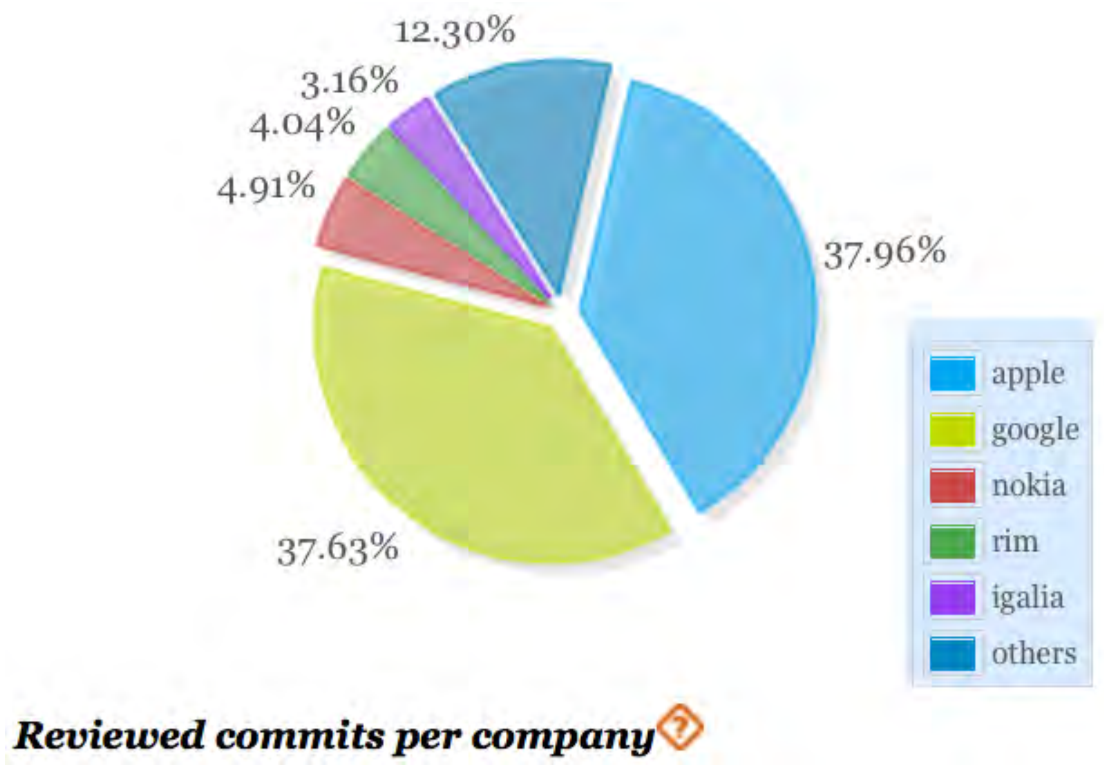
WEBKIT API

- Made up of **two** major components.
- **JavaScriptCore - responsible for everything JavaScript:**
  - JavaScript/JSON parsing & execution
  - Garbage collection
  - Debugger
  - Etc.
- **WebCore – responsible for everything else:**
  - Resource loading
  - Content parsing & rendering
  - Web Inspector
  - Etc.



KNOWN IMPLEMENTATIONS & FORKS

Image credit: http://bitergia.com/public/reports/webkit/2013_01/



- Apple's Safari
- Android's web browser
- Nokia QT
- JavaFX WebView
- WebKitGTK+
- PhantomJS
- Google Chromium
- Node WebKit
- Many more… (https://trac.webkit.org/wiki/Applications%20using%20WebKit)

WHY USE WEBKIT?

Pros

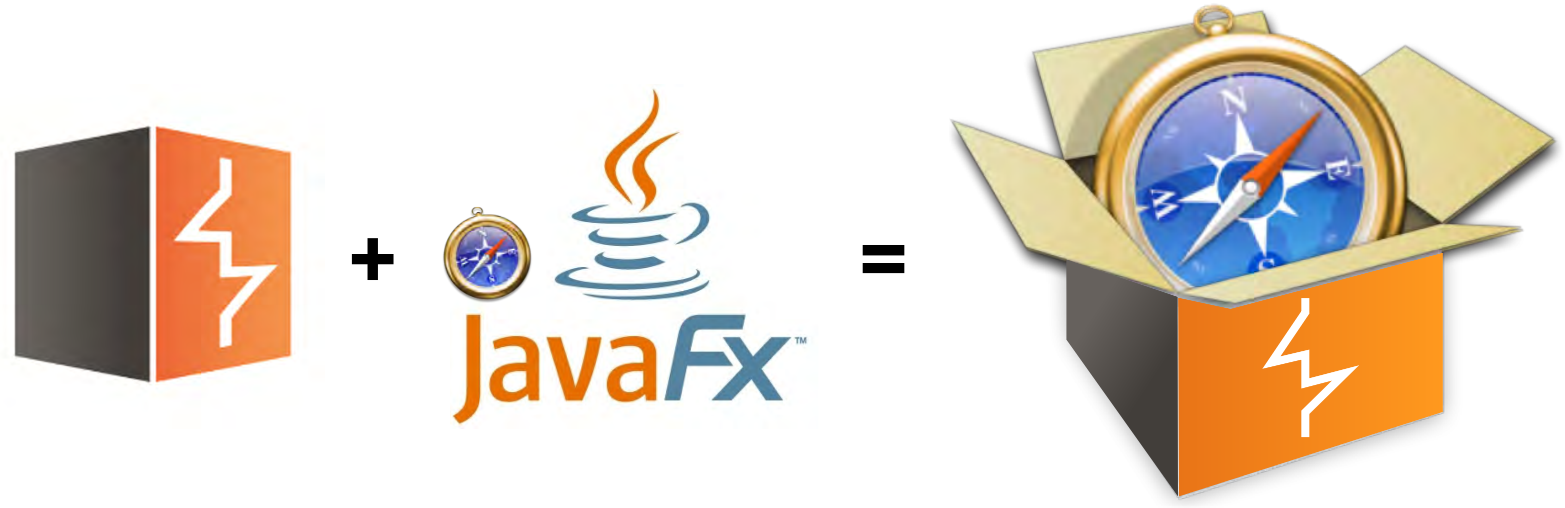
- ✓ Widespread adoption
- ✓ Lots of language support: Java, Python, C/C++, JavaScript, etc.
- ✓ Portable across many platforms
- ✓ Can interact with the DOM and JS Engine.

Cons

- ✗ Your code will be susceptible to the same bugs that plague modern browsers
- ✗ Tools will be hungrier for system resources (i.e. RAM, CPU).

HOW CAN YOU USE WEBKIT?

Language	***Libraries***
JavaScript (NodeJS)	Node WebKit
Python	WebKitGTK+, PyQt
JAVA	FX WebView, Qt Jambi, JxBrowser
Swift/ObjC	UIWebView
Ruby	WebKitGTK+, Qt
C/C++	Chromium, WebKit

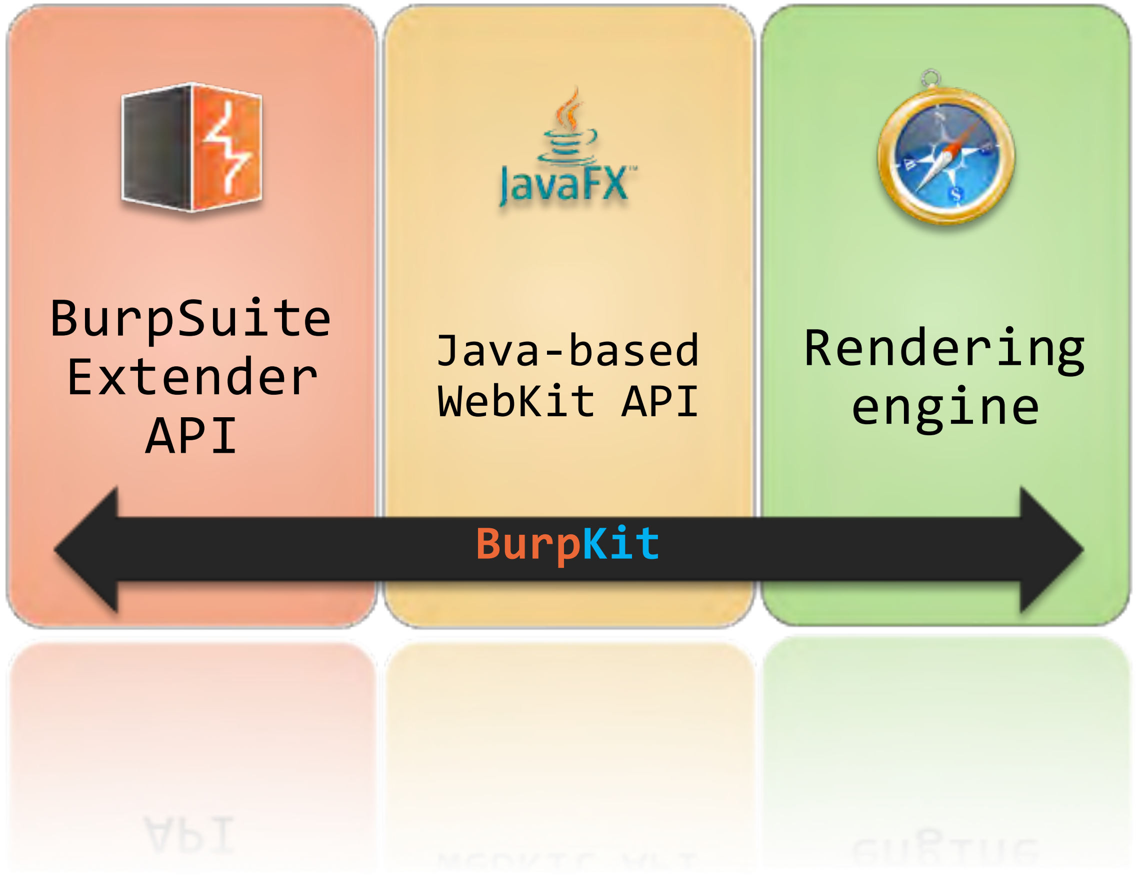


BURPKIT

How we used WebKit.

WHAT IS BURPKIT?

- **BurpKit** = **Burp**Suite + Web**Kit**
- Used JavaFX's implementation of WebKit
  - WebView & Debugger
  - WebEngine
- Provides a ***real*** rendering tab (that's right… no more lobo)
- Has a bidirectional bridge between BurpSuite & WebKit!
- And more!



DESIGN DECISIONS

- Chose to go with **JavaFX** over **JxBrowser** – why?
- **Redistribution:**
  - JavaFX comes with Java **1.8+**.
  - JxBrowser needs bundling (**>250MB**)
- **Cost:**
  - JavaFX is **FREE!**
  - JxBrowser is **not!**
- **API:**
  - JavaFX has a cleaner API
  - JxBrowser's is a bit ¿clunky?

JAVAFX: PROS AND CONS

Pros

✓Easy-to-use & clean API

✓Complete JavaScript bridge

✓Portable across many platforms

✓Leverages the Java URL framework (hookable)

✓Does provide debugging/profiling information (with some hacking)

✓Bundled with Java 1.8+

Cons

✗ API is incomplete – under development

✗ No GUI components for WebInspector and friends

✗ Little documentation on advanced features (must look at code)

✗ Still a bit buggy

IMPLEMENTATION

Nerd Rage

CHALLENGES

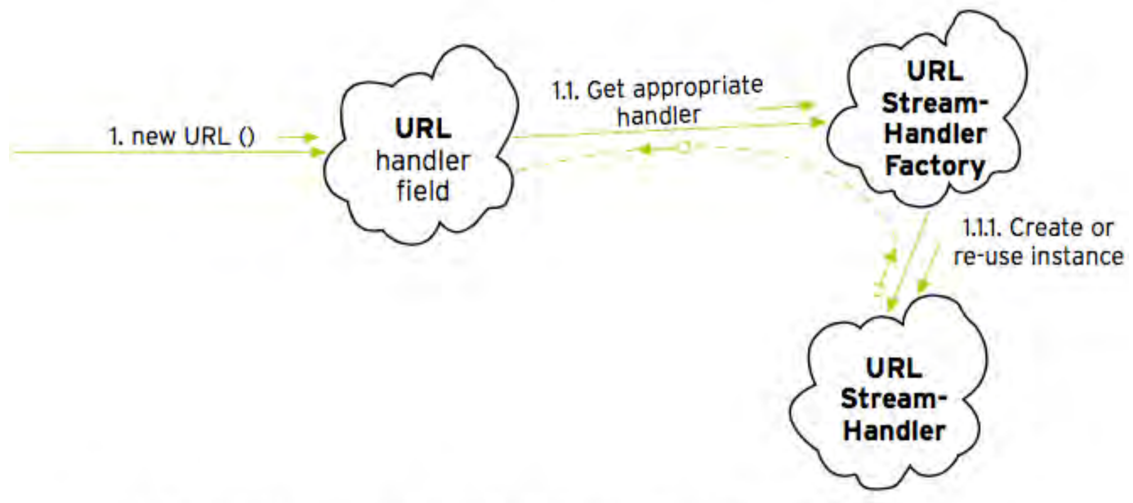
- Burp uses **Swing** for its GUI
  - `WebView` and `WebEngine` need to run on **FX** event loop
- `WebEngine` does not have a `loadContentWithBaseUrl(content, url)` method - only has:
  - `loadContent(content, type)`; and
  - `load(url)`
- **BurpSuite** had to be able to interact with **JavaScript** and vice-versa

CHALLENGE: SWING/FX INTEROP

- **Solution:** `javafx.embed.swing.JFXPanel`
- **Gotchas:**
  - Must avoid interweaving blocking calls
    - i.e. Swing → JavaFX → Swing = **¡DEADLOCK!**
  - Always check if you're on the right event loop!
- **Workarounds:**
  - Eagerly initializing resources sometimes necessary
  - Lots of wrapping code!

CHALLENGE: LOADING CONTENT WITH A BASE URL

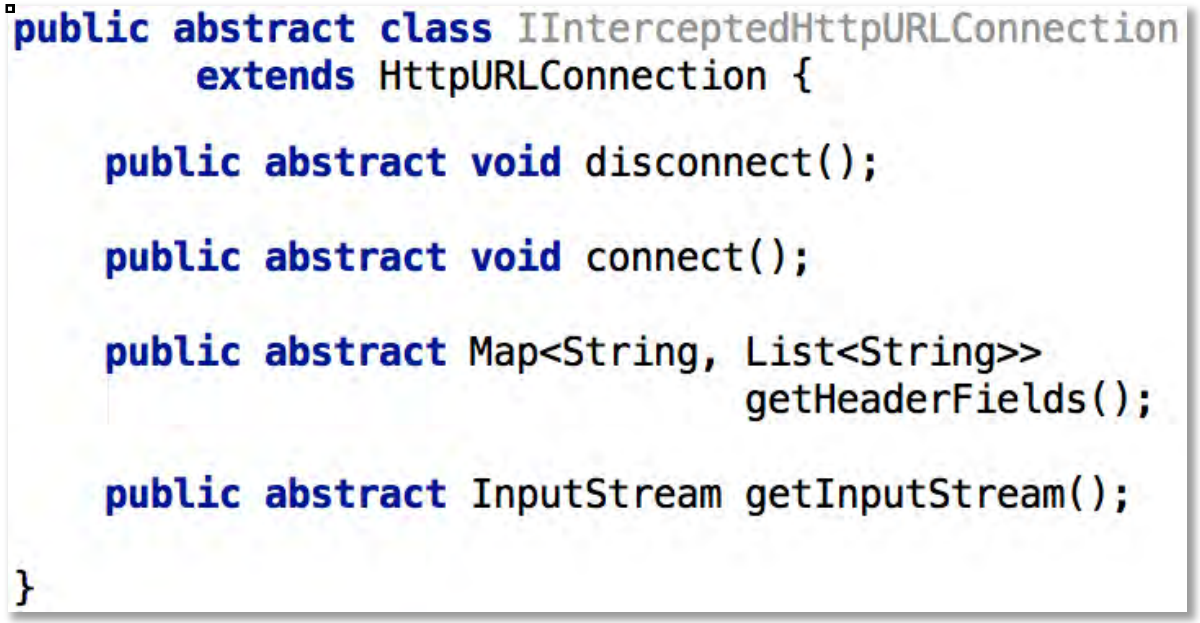
- **Why?**
  - Required to render responses for repeated requests
- **Solution:** hook `java.net.URL` protocol handling framework
  - WebView uses framework to issue HTTP(S) requests
- **Challenge:**
  - Our new handlers would have to support both live and repeated requests.


1. new URL ()
URL handler field
1.1. Get appropriate handler
URL Stream-Handler Factory
1.1.1. Create or re-use instance
URL Stream-Handler

• Figure 2: Sequence of interactions between java.net objects to resolve a URL into a stream

Credit: http://media.techtarget.com/tss/static/articles/content/dm_protocolHandlers/java_protocol.pdf

CHALLENGE: REPEATER



```
public abstract class IInterceptedHttpURLConnection
        extends HttpURLConnection {

    public abstract void disconnect();

    public abstract void connect();

    public abstract Map<String, List<String>>
                                getHeaderFields();

    public abstract InputStream getInputStream();

}
```

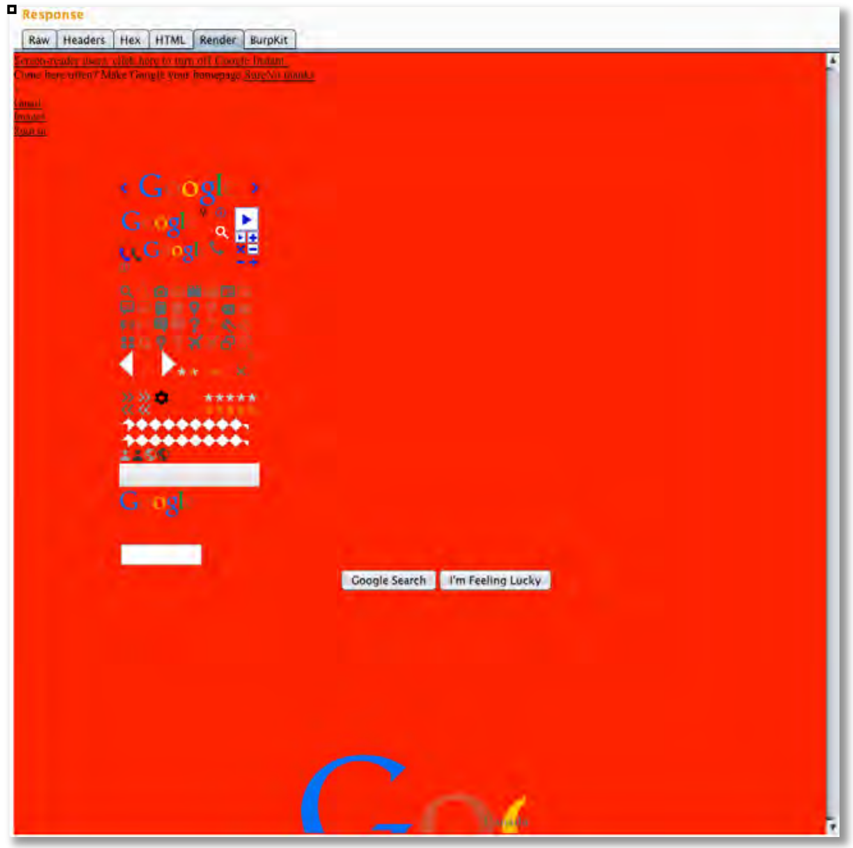
- **Background:** did not want to reissue a live request because content may change.
- **Solution:** overrode HTTP(s) handlers and used `User-Agent` to “tag” repeated requests.
  - If `User-Agent` contains SHA1 hash, give URL handler fake output stream
  - Else, continue with live request
- See BurpKit Java package **`com.redcanari.net.http`** for code.

CHALLENGE: JAVASCRIPT BRIDGE

- **Background:** need to be able to query and manipulate DOM
- **Solution:** inject **JAVA** objects into **JS** engine!
- **Gotchas:**
  - Funky reflection algorithm in `WebEngine` prevented straight-forward JAVA object interaction.
  - Lots of deadlock scenarios
- **Workarounds:**
  - Wrapper classes galore!
  - Eager instantiation of Swing components.

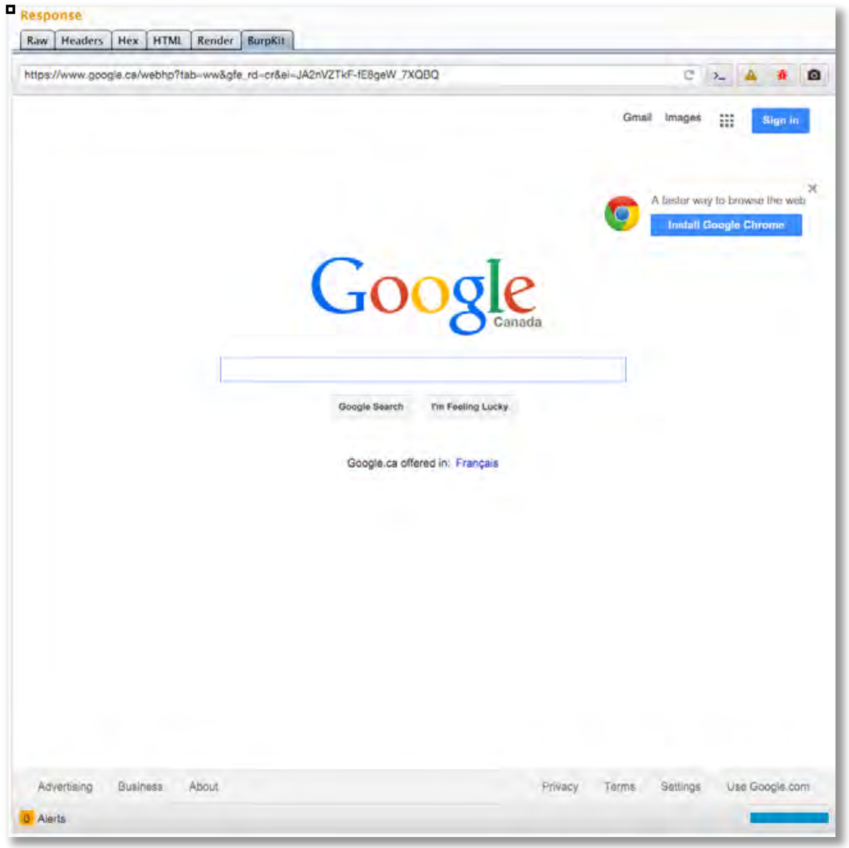
THE FINAL PRODUCT

Google: Before



Screen-reader users, click here to turn off Google Instant.
Come here often? Make Google your homepage. Sure/No thanks

& After



https://www.google.ca/webhp?tab=ww&gfe_rd=cr&ei=JA2nVZTkF-fE8geW_7XQBQ

A better way to browse the web

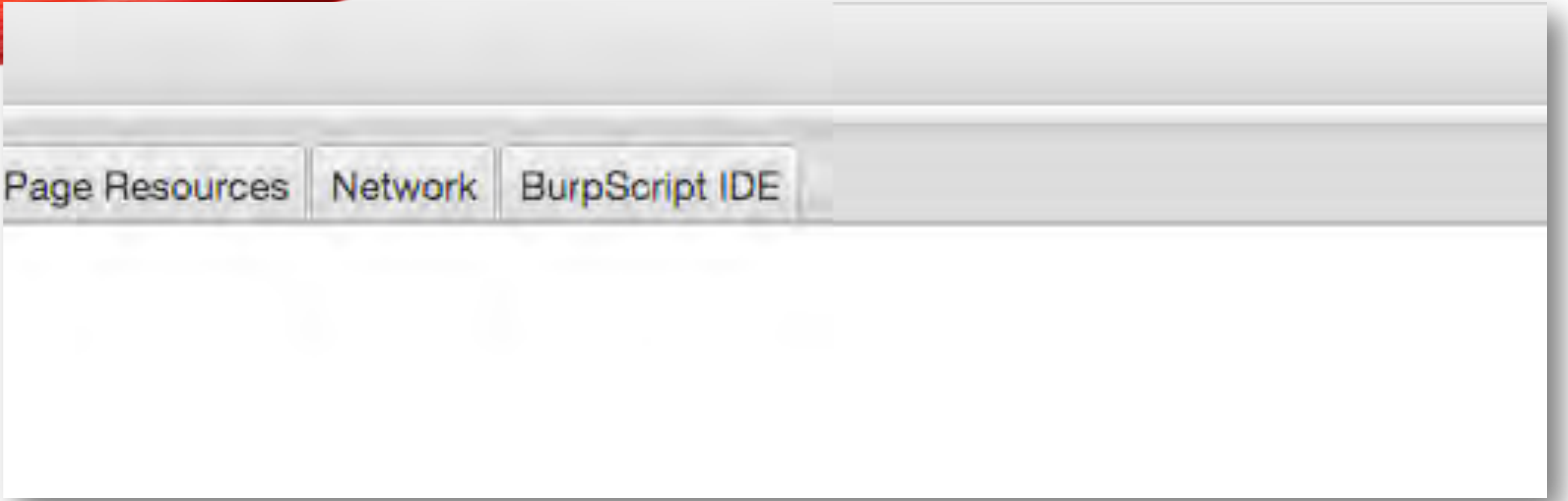
Google.ca offered in: Français

WELL?

"when something exceeds your ability to understand how it works, it sort of becomes magical"
Jonathan Ive

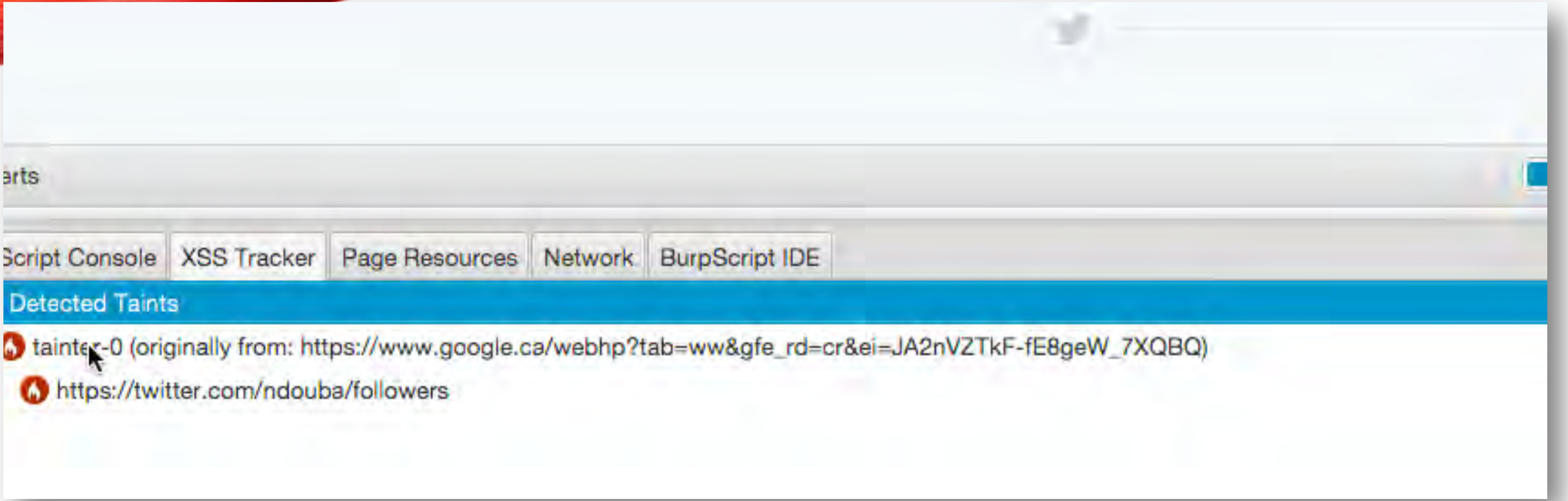
BURPKIT DEMOS

There's lots to see!



DEMO: GUI WALKTHROUGH

Feature set

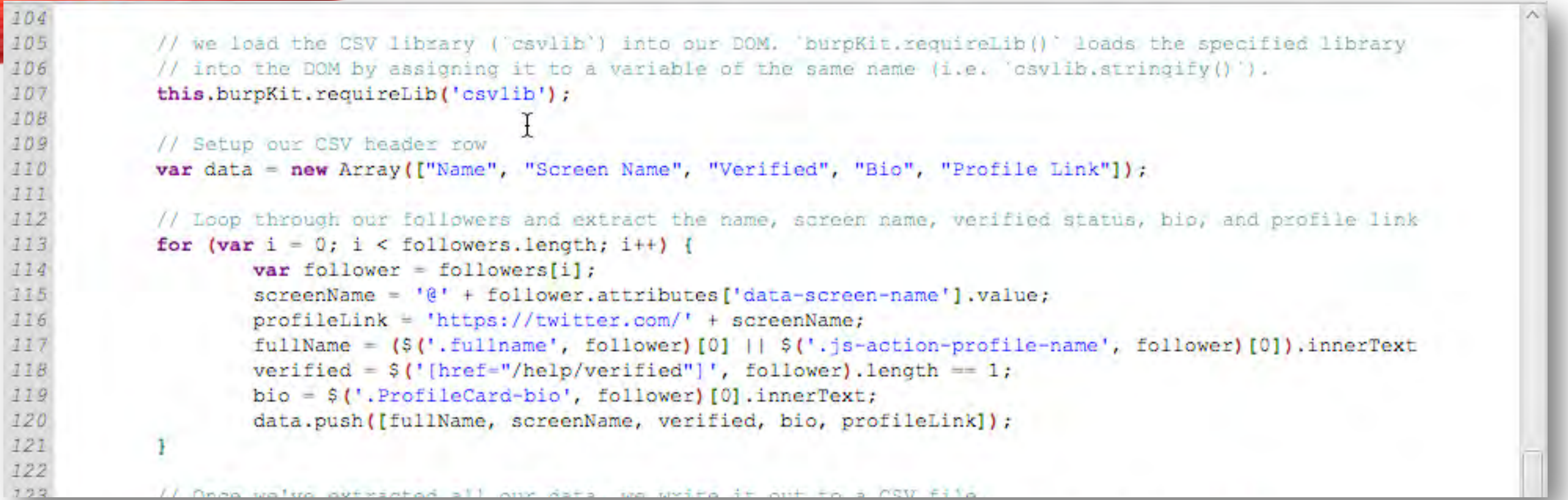


tainter-0 (originally from: https://www.google.ca/webhp?tab=ww&gfe_rd=cr&ei=JA2nVZTkF-fE8geW_7XQBQ)

https://twitter.com/ndouba/followers

XSS TRACKER

Tainting applications

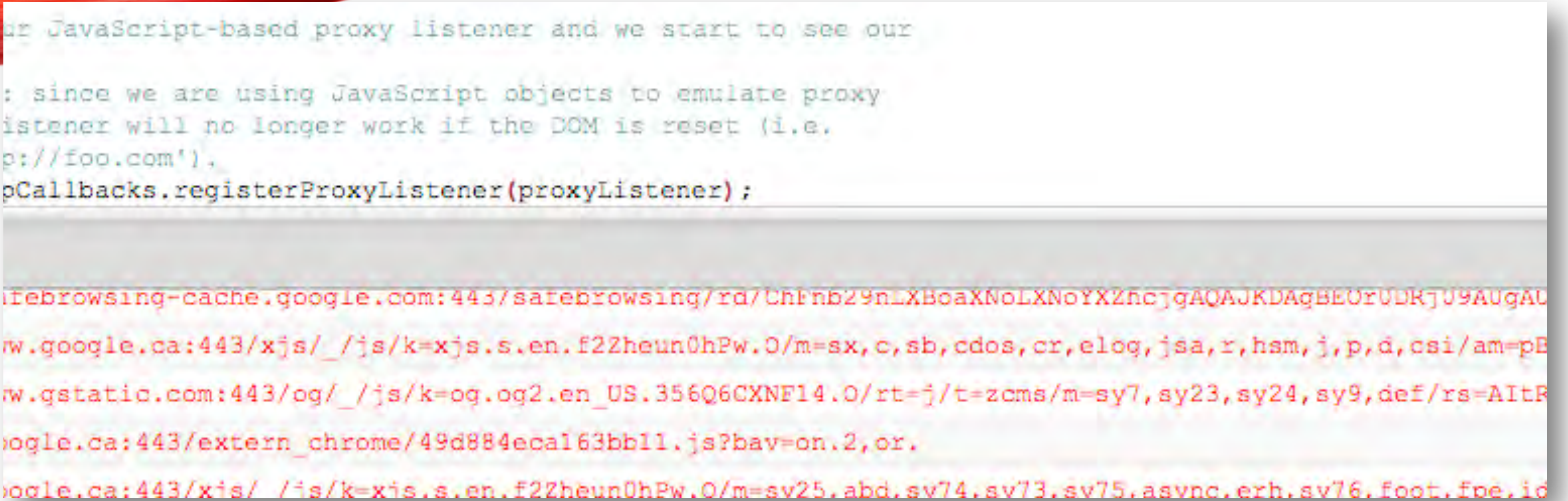


```

        // we load the CSV library (`csvlib`) into our DOM. `burpKit.requireLib()` loads the specified library
        // into the DOM by assigning it to a variable of the same name (i.e. `csvlib.stringify()`).
        this.burpKit.requireLib('csvlib');

        // Setup our CSV header row
        var data = new Array(["Name", "Screen Name", "Verified", "Bio", "Profile Link"]);

        // Loop through our followers and extract the name, screen name, verified status, bio, and profile link
        for (var i = 0; i < followers.length; i++) {
                var follower = followers[i];
                screenName = '@' + follower.attributes['data-screen-name'].value;
                profileLink = 'https://twitter.com/' + screenName;
                fullName = ($('.fullname', follower)[0] || $('.js-action-profile-name', follower)[0]).innerText
                verified = $('[href="/help/verified"]', follower).length == 1;
                bio = $('.ProfileCard-bio', follower)[0].innerText;
                data.push([fullName, screenName, verified, bio, profileLink]);
        }

        // Once we've extracted all our data, we write it out to a CSV file.
```

DEMO: DOM INTERACTION

Analyzing Twitter Followers

redcanari INFORMATION SECURITY



```
ur JavaScript-based proxy listener and we start to see our

: since we are using JavaScript objects to emulate proxy
istener will no longer work if the DOM is reset (i.e.
p://foo.com').
pCallbacks.registerProxyListener(proxyListener);
```

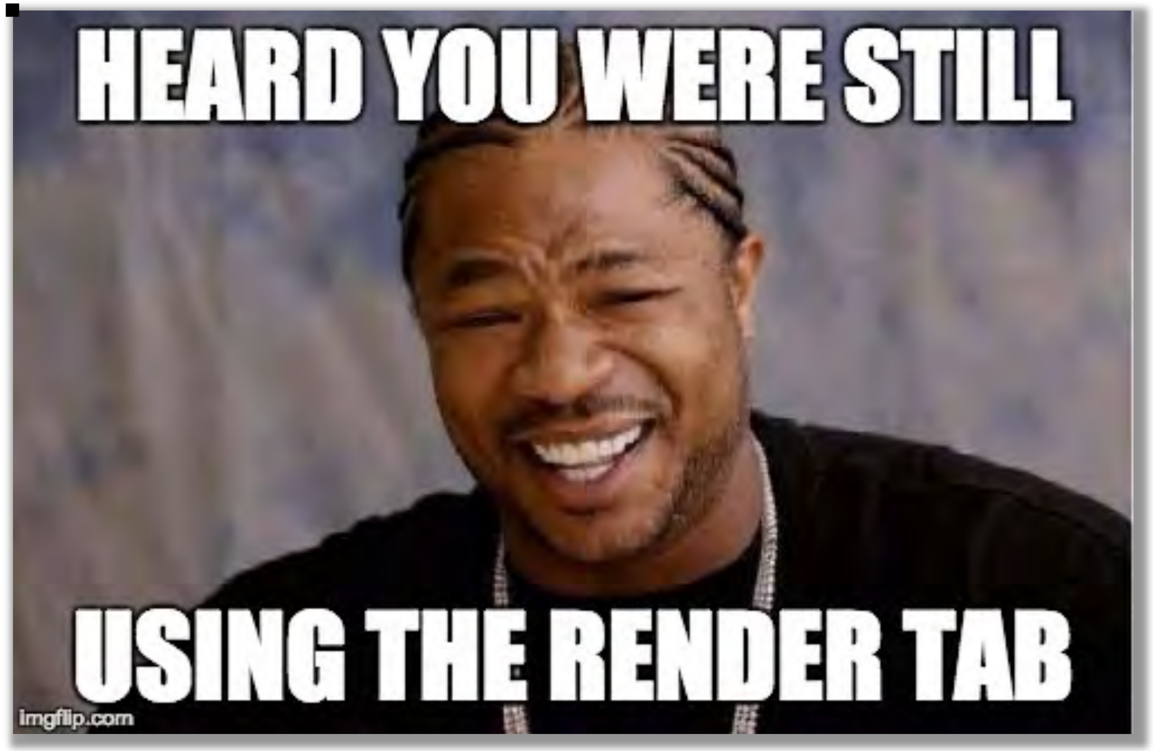
```
ifebrowsing-cache.google.com:443/safebrowsing/rd/ChFnb29nLXBoaXNoLXNoYXZhcjgAQAJKDAgBEOrUDRjU9AUgAU
w.google.ca:443/xjs/_/js/k=xjs.s.en.f2Zheun0hPw.O/m=sx,c,sb,cdos,cr,elog,jsa,r,hsm,j,p,d,csi/am=pB
w.gstatic.com:443/og/_/js/k=og.og2.en_US.356Q6CXNF14.O/rt=j/t=zcms/m=sy7,sy23,sy24,sy9,def/rs=AItR
ogle.ca:443/extern_chrome/49d884eca163bb11.js?bav=on.2,or.
ogle.ca:443/xjs/_/js/k=xjs.s.en.f2Zheun0hPw.O/m=sy25,abd,sy74,sy73,sy75,async,erh,sy76,foot,fpe,id
```

DEMO: BURP EXTENSIONS

Proxy Listeners, Message Editors, and Context Menus

redcanari INFORMATION SECURITY

CONCLUSION



- Let's stop scraping and let's start **DOM**inating the web!
- Our security tools need to evolve just like the web.
  - We have the tools/libraries at our disposal
- Please contribute your ideas and code to BurpKit!
  - We need to make it the standard!

KUDOS

- My ***Lovely*** Wife ☺
- Justin Seitz
  - http://automatingosint.com/
- Dirk Lemmermann
  - http://dlsc.com/
- Tomas Mikula
  - https://github.com/TomasMikula/RichTextFX
- Java/JavaFX team
- The Noun Project
- All the contributors!

redcanari INFORMATION SECURITY

¿QUESTIONS?

We aim to please…