Next Generation Clickjacking

New attacks against framed web pages

Black Hat Europe, 14th April 2010

Paul Stone
paul.stone@contextis.co.uk
Coming Up...

- Quick Introduction to Clickjacking
- Four New Cross-Browser Attack Techniques
- Clickjacking Tool
- Browser Specific Exploits
Clickjacking in 60 seconds
A quick recap
It’s all about iframes

Any site can frame any other site, even https

<iframe src="https://www.google.com/..."></iframe>
Iframes

Same-origin policy access prevents JavaScript access to content from another domain.

```
http://attacker.com

frames[0].document
"Access is denied.

frames[0].location.href
"Permission denied"
```
**Step 1 – Frame Content**

```html
<iframe src="http://mobile.twitter.com?status=Don't click this..." width="600" height="300">
```

![Twitter inner.html](inner.html)
Step 2 – Position Target

```html
style="position: absolute; left: -15px; top: -106px"
```
Step 3 – Crop and Position Target

<iframe src="inner.html" width="100" height="25">

Fully visible iframe

iframe layered over button

iframe following cursor

</iframe>
Clickjacking vs. The Rest
Browser based attacks compared
Clickjacking vs. The Rest

Cross-Site Scripting
• 2,700,000 Google Results  ~14 years old

Cross-Site Request Forgery
• 542,000 Google Results  ~10 years old

Clickjacking
• 281,000 Google Results  ~2 years old

All allow a malicious website to interact with web sites you’re logged in to.
Cross-Site Scripting (XSS)

- Inject JavaScript into a web application

http://mymail.com/search?foo<script>doBadStuff()</script>

- Gives an attacker control of a user’s session and data
  - **Read** user data (e.g. emails, documents)
  - **Execute** commands and **inject** data (e.g. transfer money in online banking)
  - Little to no user interaction required
  - Thwarted by correct output escaping:

    &lt;script&gt;doBadStuff()&lt;/script&gt;
Cross-Site Request Forgery (CSRF)

- Trick a web application into honouring requests sent from a malicious web site

http://mybank.com/transfer?amt=10000&acct=badguy

- Allows an attacker to perform actions as user
  - A write only attack; cannot read back results
  - Little to no user interaction required
  - Thwarted by adding a random token to requests

http://mybank.com/transfer?amt=50&acct=friend&token=e43d2af7ecb
Clickjacking

- Get user to click on stuff in hidden frame
- Flash Webcam/Microphone Access (fixed)

- Allows an attacker to perform actions as user
  - Bypass CSRF protection
  - Can only inject **clicks**, not data
  - Can break if page layout changes
  - More user interaction required
  - Thwarted by anti-framing:

X-Frame-Options
If (top !== window) top.location = window.location.href;
Clickjacking + CSRF

• Can’t do CSRF due to random token:
  POST /status/update HTTP/1.1
  Host: twitter.com
  Cookie: _twitter_sess=xxx;

  authenticity_token=r4nD0Mt0k3n&status=hello

• So prime form with data using ‘partial CSRF’ (or Twitter feature)
  http://twitter.com?status=hello

• Use hijacked click to submit form
**Bugzilla CSRF Protection**

You submitted changes to process_bug.cgi with an invalid token, which may indicate that someone tried to abuse you, for instance by making you click on a URL which redirected you here *without your consent*.

Are you sure you want to commit these changes?

![Yes, Confirm Changes button]

---

CSRF Protection can make clickjacking simple
Better Target Positioning

or, the problem with pixels
The Problem with Pixels

When the attack is prepared…
The Problem with Pixels

When the attack is executed... Oops!
Fragments and Anchors

- Remember this - `<a name="subheading">` http://example.org/page.html#subheading

- Browser will scroll to anchor element

- Also works with **any** ID attribute:
  `<input type="submit" value="Save" id="wpSave">`

action=edit#wpSave
Element IDs

First name:
Last name:
Nickname:
Zip code: (optional)
Country: (optional)
Time zone:

Add an alternate email address to your account

You can use alternate email addresses to sign in to your Google Account, recover your password, and more. Alternate email addresses can only be associated with one Google Account at a time.

Note: In some Google services, if you share your alternate email address with your contacts, they might be able to learn your primary email address.

Email

Add an additional email address:

Save    Cancel

Element IDs on Google Accounts page
Fragment Positioning

- Works with nested frames
- Browsers will scroll horizontally + vertically to make target visible
- Can do relative positioning:

```javascript
innerFrame.src = targetUrl + '#fragment';
outerFrame.scrollBy(100, 20);
```

- Demo
Technique #1 – Text Field Injection
Bypassing CSRF more effectively
Drag and Drop Data Transfer

- All browsers implement Drag and Drop API
- First in IE, now part of HTML 5
- Can drag data across domains

<div ondragstart="event.dataTransfer.setData('some text')">Drag me</div>
Drag and Drop Clickjacking

1. Position text field in hidden iframe
2. Get user to start dragging something
   • Scrollbar, slider, game piece
3. Set drag data
4. Make iframe follow cursor
5. User releases mouse button, drops text into field
6. Position submit button in iframe
7. Get user to click
Frog. Blender. You know what to do
Text Field Injection

- One drag per text field (not ideal)
- Completely bypass CSRF
- Could be used to target webmail, document editors
- Works in latest IE, Firefox, Safari, Chrome
Technique #2 – Content Extraction

Beyond CSRF
Content Extraction

- Reverse drag and drop
- Drag content from inside iframe
- `<body ondrop="alert(event.dataTransfer.getData('Text'))">`
- Links and images are draggable as URLs
- Links may contain sensitive information
  - Hashes, Object IDs, User information

https://docs.google.com/Doc?docid=0Acwo2Bn17-PrZGZudHRobnJfNDNmOTZzOTIkbg&hl=en_GB

- …but we can do better
Text Selections

• Selections are draggable
• Can we get a user to select text on a page?
• - and then drag selection onto attacker’s page?
• Not as tricky as it sounds…
1. Position target A in iframe
2. User starts to drag
3. Position target B in iframe
4. User finishes drag
5. Position target C
6. User starts to drag
7. .. and drops on attacker's page
Content Extraction – HTML Source

- If we can get the HTML source of a page, we get:
  - URLs for every link on page
  - Source code of inline JavaScript
  - Values in hidden form fields
  - ‘Secret’ values such as CSRF tokens

- Use editable HTML area as drop target
  - designMode or contentEditable area
Content Extraction

- Two drag and drops needed for each page
- Position doesn’t matter!
- Could be used for intranet reconnaissance
- Works in latest IE, Firefox, Safari, Chrome
  - But no hidden form fields or script tags in WebKit browsers
Technique #3 – Java Drag and Drop

More fun with text injection
Forced Drag and Drop

- Java DnD API available in Java applets
- Can extend MouseDragGestureRecognizer class
- Trigger drag from a click

- JavaScript can call applet to trigger drag at any time
  - …even if mouse is not over applet
  - …even if mouse button is not held down
- Text is dropped onto element under mouse
Form Spraying

- Fill many form fields in one go
  1. Position text field in iframe under mouse
  2. Force drop of text into field
  3. Repeat for each field
  4. Click to Submit

- Details vary between browsers and platforms
  - Chrome requires mouse movement between each drop
  - Works on Windows and MacOS X but not Linux
Technique #4 – Leaky Iframes
Login detection and much more
Fragment Positioning

- Browser will scroll iframe to make element visible
- Clickjacking uses big inner iframe, small outer iframe
- Outer iframe is scrolled
- Outer iframe is controlled by attacker
- Attacker can read scroll position
Fragment Position Detection

- Load page in inner iframe
- Make outer iframe tiny (10x10)
- Navigate to URL + #fragment
- Read scroll position of outer iframe

- If position didn’t change, element is not on page
- If it did, we know there’s an element with that ID and where it is on the page
What can we do

- More targeted attacks
- Check for login page IDs
  - Is a user logged into site X
  - Check if ‘re-authentication’ is needed for sensitive pages
- Check position of page footer
  - How many emails in your inbox
  - How many results for search query X
- Brute force numeric IDs
  - What items in your shopping cart / order history
  - This is quick as page doesn’t reload if only #fragment changes in URL
Clickjacking Countermeasures
and how to break them
How to Protect against Clickjacking

• Don’t allow your site to be framed
• Use X-Frame-Options and JavaScript

• X-Frame-Options only works in some browsers:
  • IE8+
  • Safari 4+
  • Chrome 2+

• Firefox will support X-Frame-Options and Content Security Policy (CSP) in a future release
How to Protect against Clickjacking

- JavaScript protection
  - Framebusting
  - Hide or obscure content
- Facebook and Twitter use JavaScript protection but not X-Frame-Options
- JavaScript protection is not 100% effective
- Even if it was, most sites still have unprotected areas
Breaking JavaScript Protection

• Prevent framebusting using 204 redirects
  http://coderrr.wordpress.com/2009/02/13/
  preventing-frame-busting-and-click-jacking-ui-redressing/

• Firefox
  • Disable JavaScript using Iframe inside designMode
  • view-source: pseudo-protocol

• Internet Explorer
  • Disable JavaScript by loading site in designMode mode
Vulnerable Sites

- High profile sites have implemented Clickjacking protection
- Most are still vulnerable through
- Mobile sites
- Gadgets / widgets allow framing
Twitter

http://mobile.twitter.com  http://m.twitter.com
http://m.facebook.com
https://mail.google.com/mail/ig/mailmax
Clickjacking Tool
point + shoot clickjacking
Clickjacking Tool

- Browser based tool
- Use all new techniques
- Position click targets visually
- Multistep attacks are easy

Latest version at: http://www.contextis.co.uk
Browser Specific Vulnerabilities
CVE-2010-0494

- ‘HTML Element Cross-Domain Vulnerability’
- aka Universal Cross-Site Scripting
- Fixed as part of MS10-018

Allows XSS on any site by forced drag and drop of HTML into an editable iframe (only mouseover required)
CVE-2010-0178

- ‘Chrome privilege escalation via forced URL drag and drop’
- Fixed in Firefox 3.6.2

Allows arbitrary code execution with just one click using forced drag and drop