

Next Generation Clickjacking

New attacks against framed web pages

Black Hat Europe, 14th April 2010

Paul Stone

paul.stone@contextis.co.uk

Coming Up...

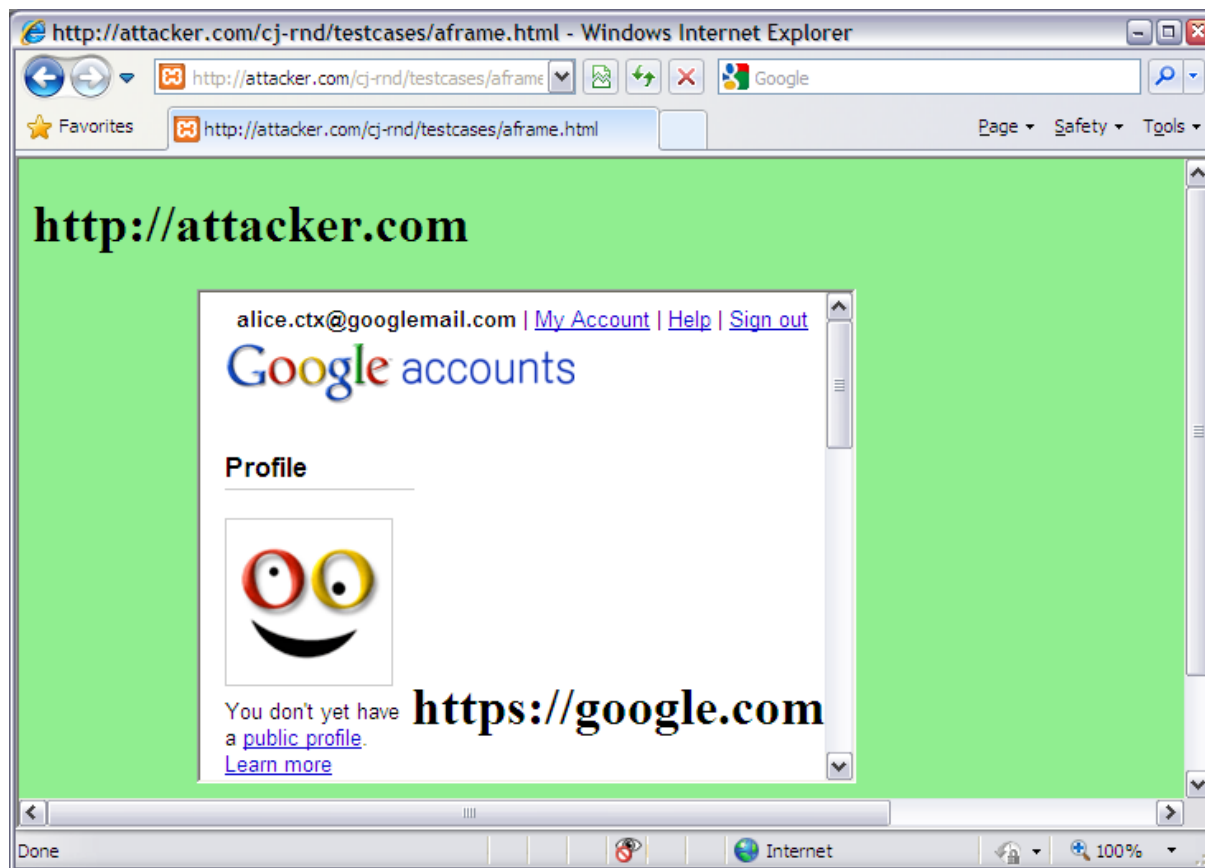
- Quick Introduction to Clickjacking
- Four New Cross-Browser Attack Techniques
- Clickjacking Tool
- Browser Specific Exploits

Clickjacking in 60 seconds

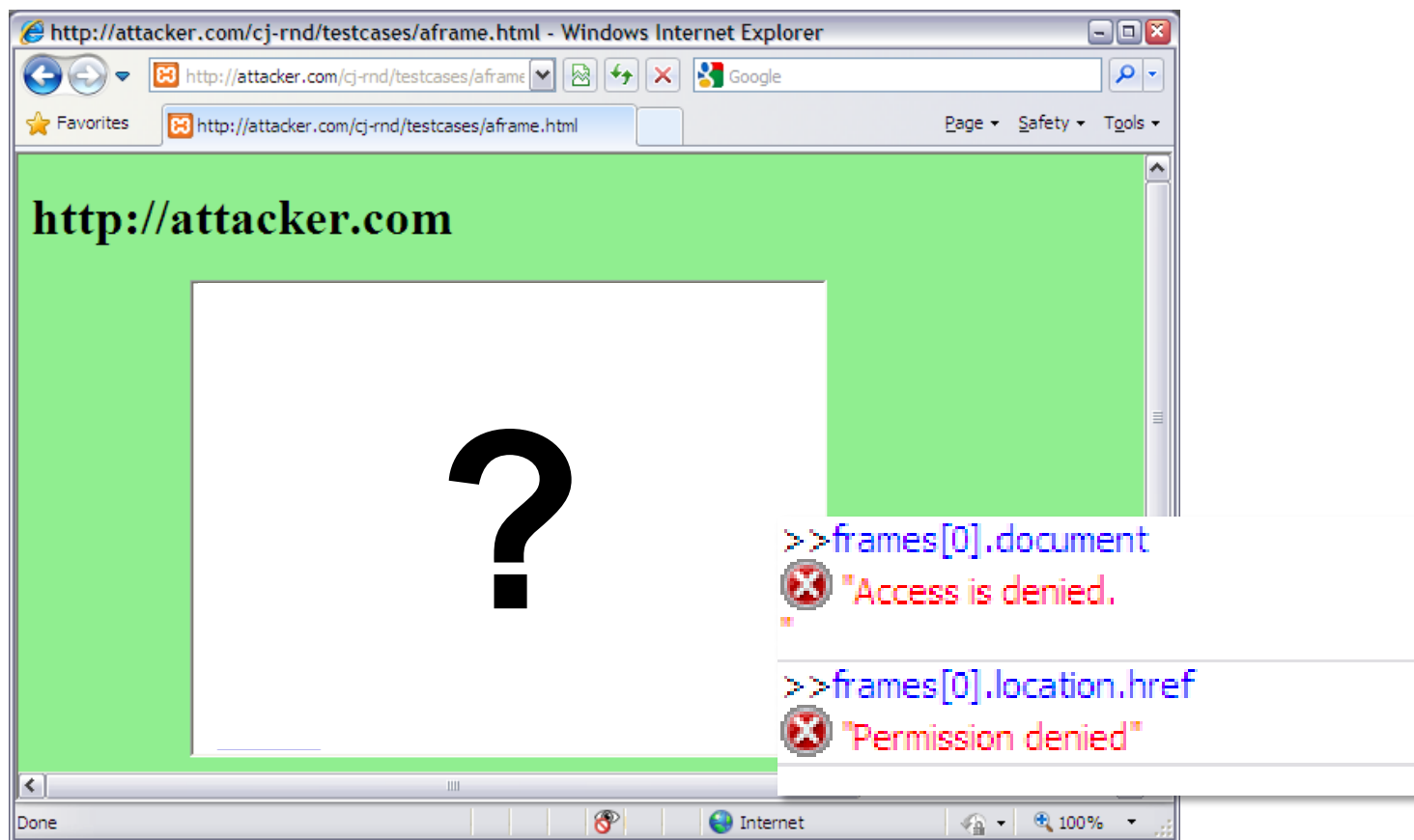
A quick recap

It's all about iframes

Any site can frame any other site, even https
<iframe src="https://www.google.com/..."></iframe>

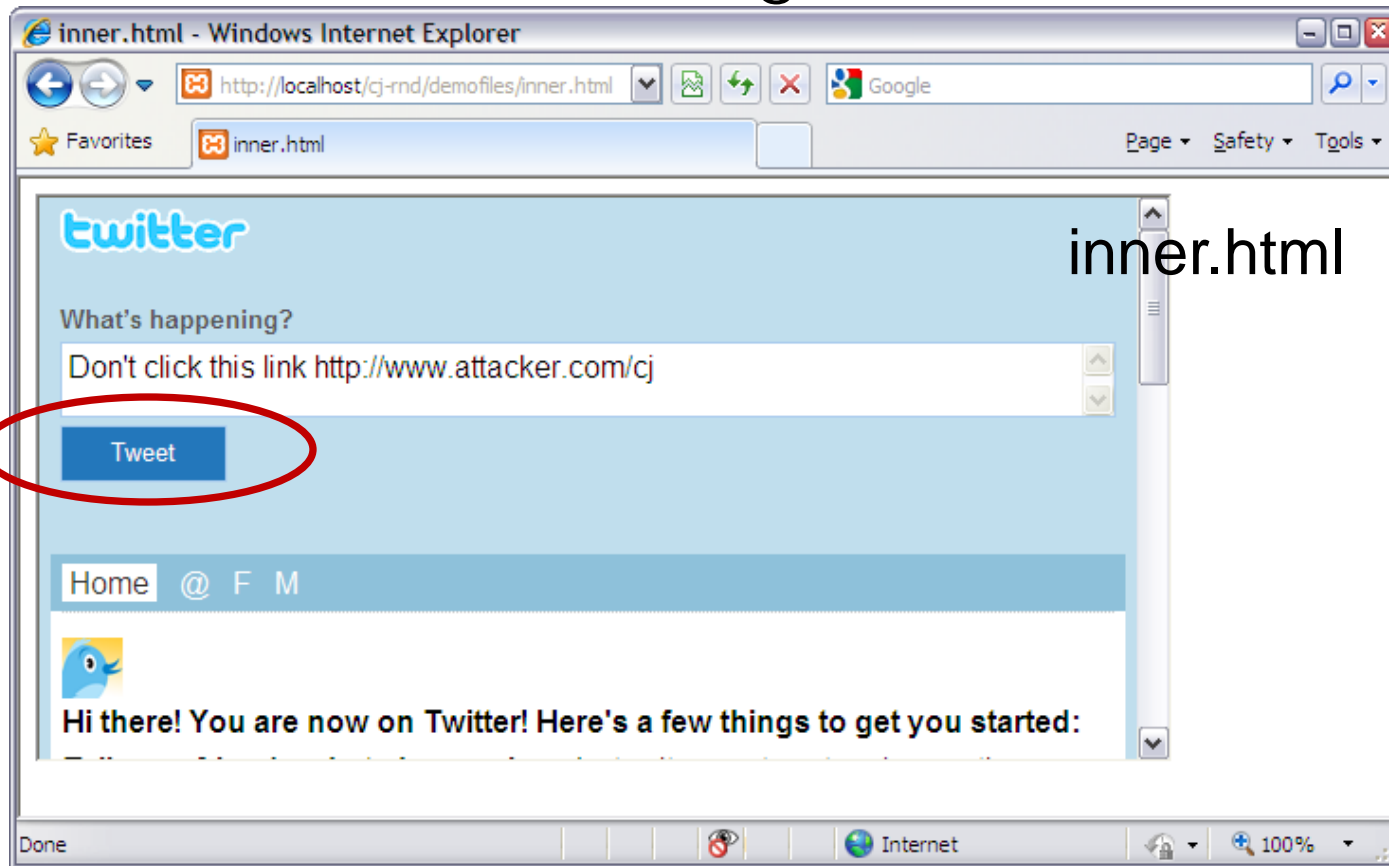


Same-origin policy access prevents JavaScript access to content from another domain



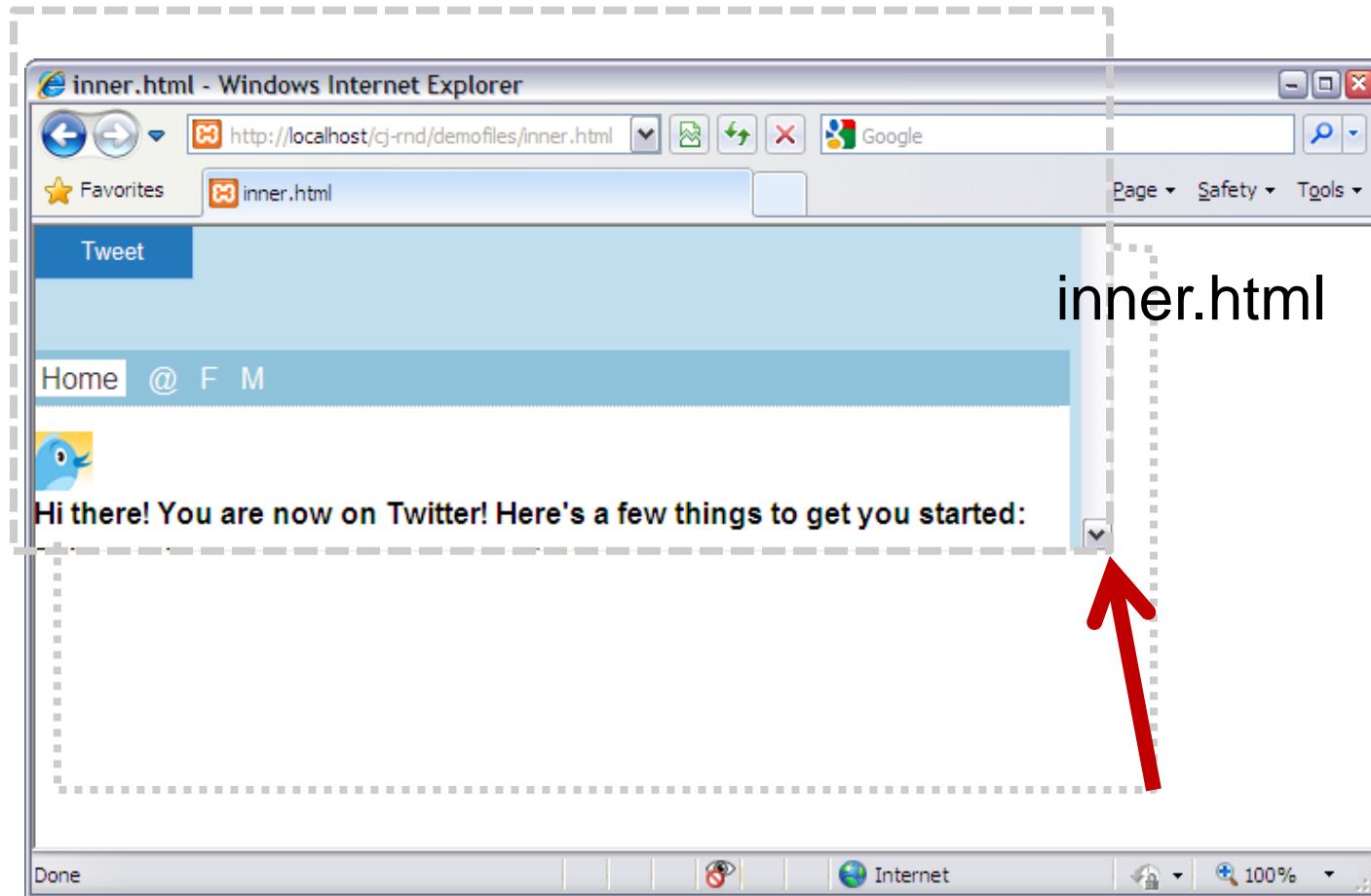
Step 1 – Frame Content

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



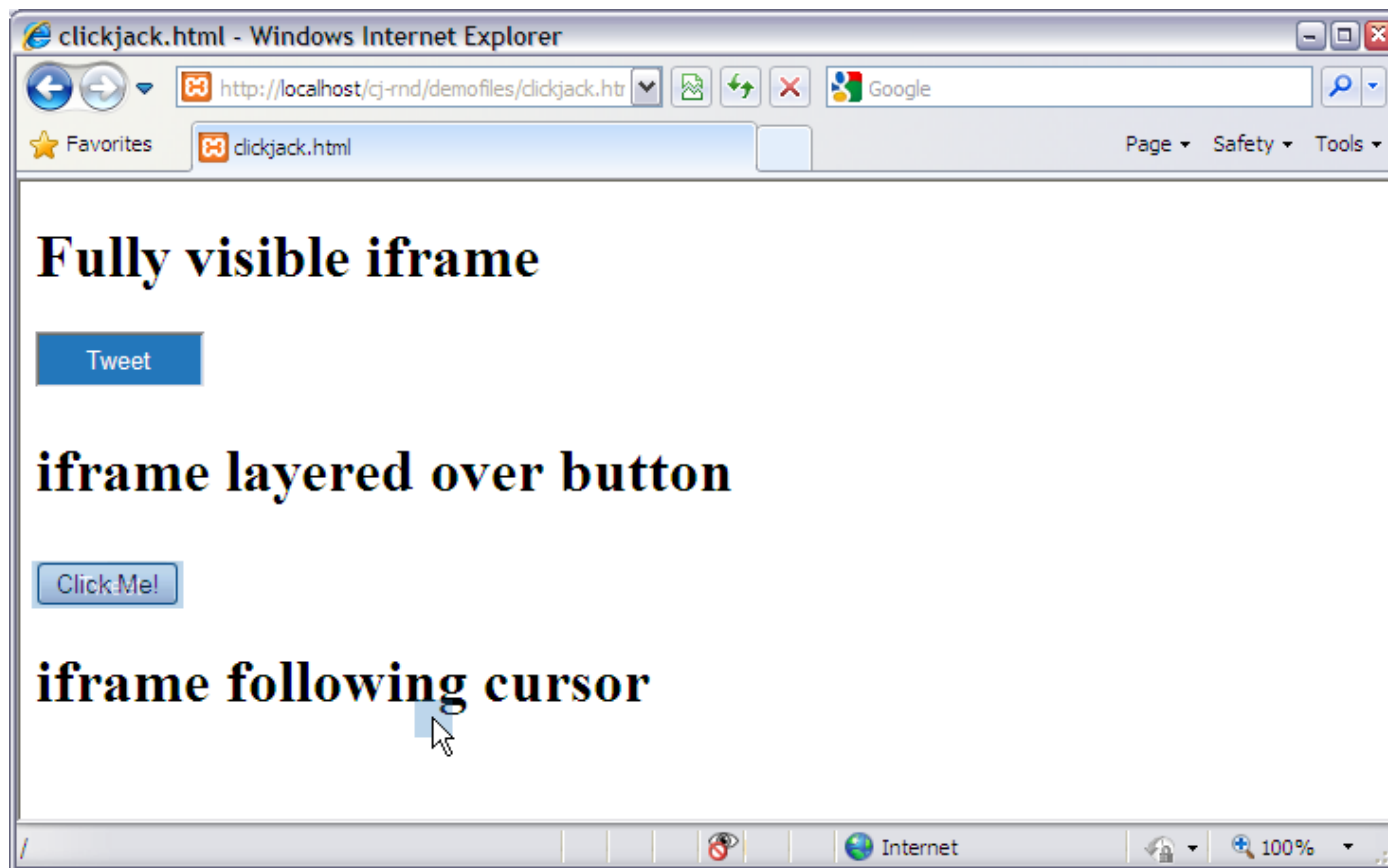
Step 2 – Position Target

style="position: absolute; left: -15px; top: -106px"



Step 3 – Crop and Position Target

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



Clickjacking vs. The Rest

Browser based attacks compared

Cross-Site Scripting

- 2,700,000 Google Results ~14 years old

Cross-Site Request Forgery

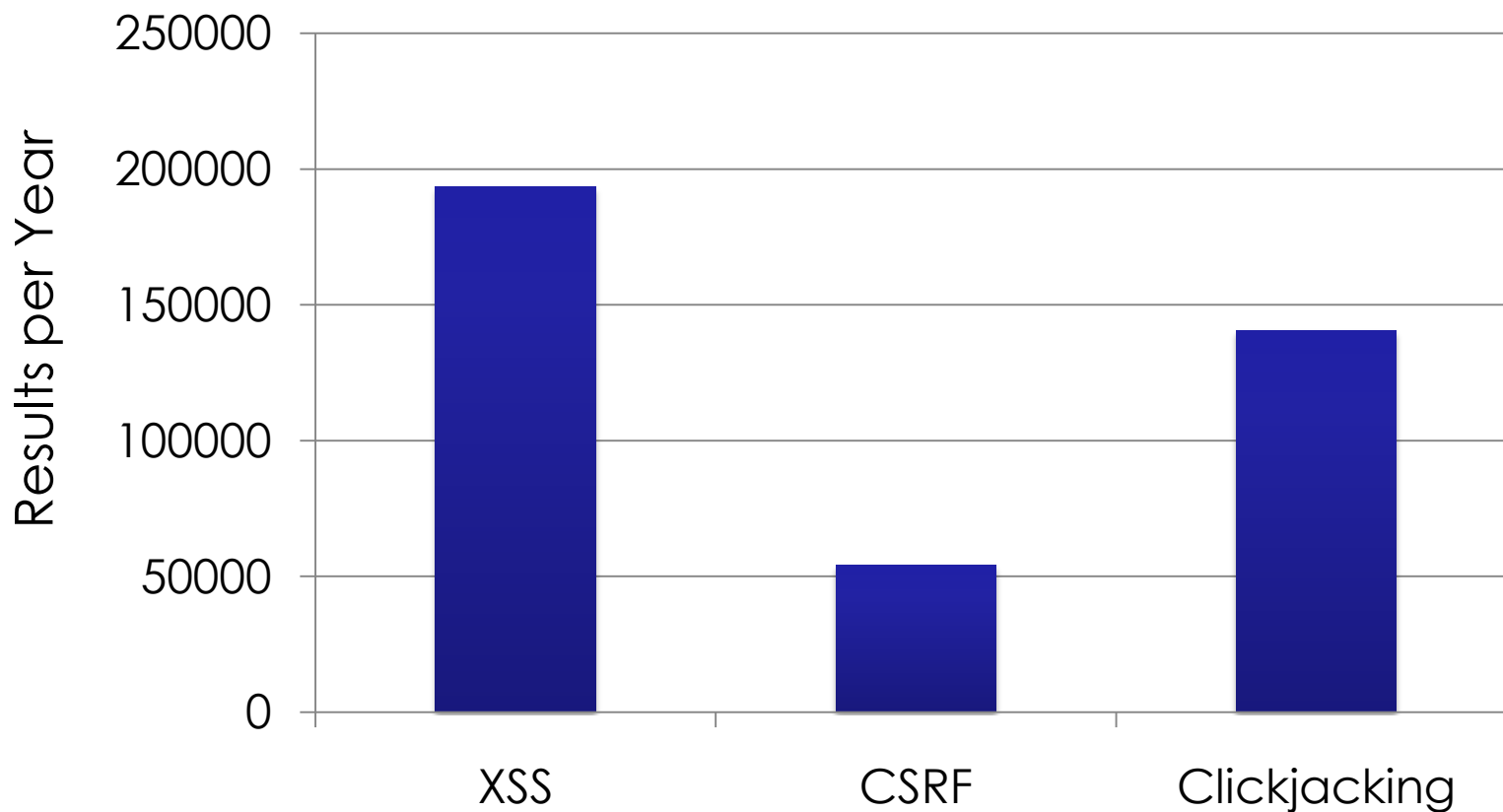
- 542,000 Google Results ~10 years old

Clickjacking

- 281,000 Google Results ~2 years old

All allow a malicious website to interact with web sites you're logged in to.

Completely Meaningless Chart



- 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:

<script>doBadStuff()<script>

- 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`

- 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;

- 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@Mozilla – Suspicious Action

[Home](#) | [New](#) | [Search](#) | | [Reports](#) | [My Requests](#) | [My Votes](#) | [Preferences](#)
| [Log out](#)

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?

[No, throw away these changes](#) (you will be redirected to the home page).

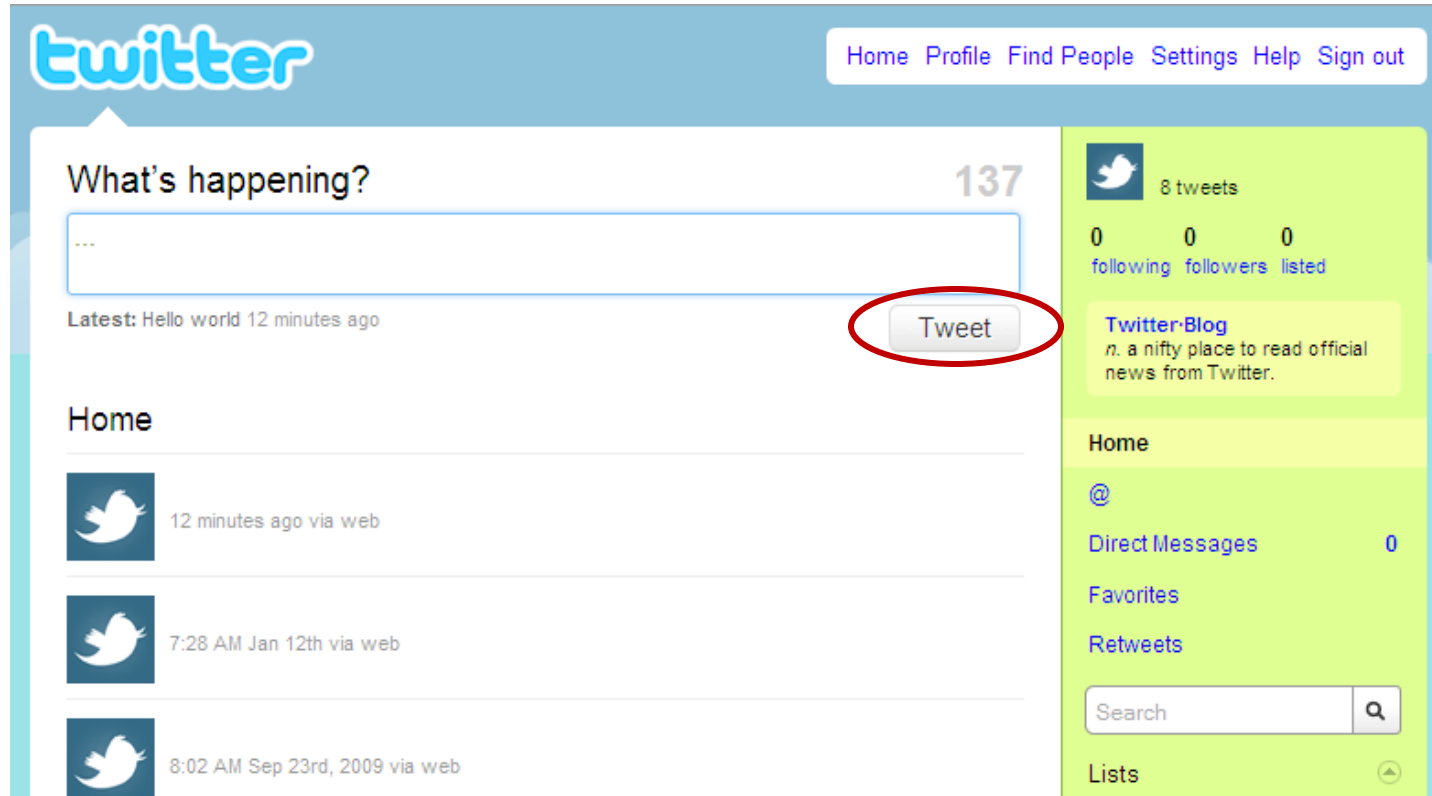
[Home](#) | [New](#) | [Search](#) | | [Reports](#) | [My Requests](#) | [My Votes](#) | [Preferences](#)

CSRF Protection can make clickjacking simple

Better Target Positioning

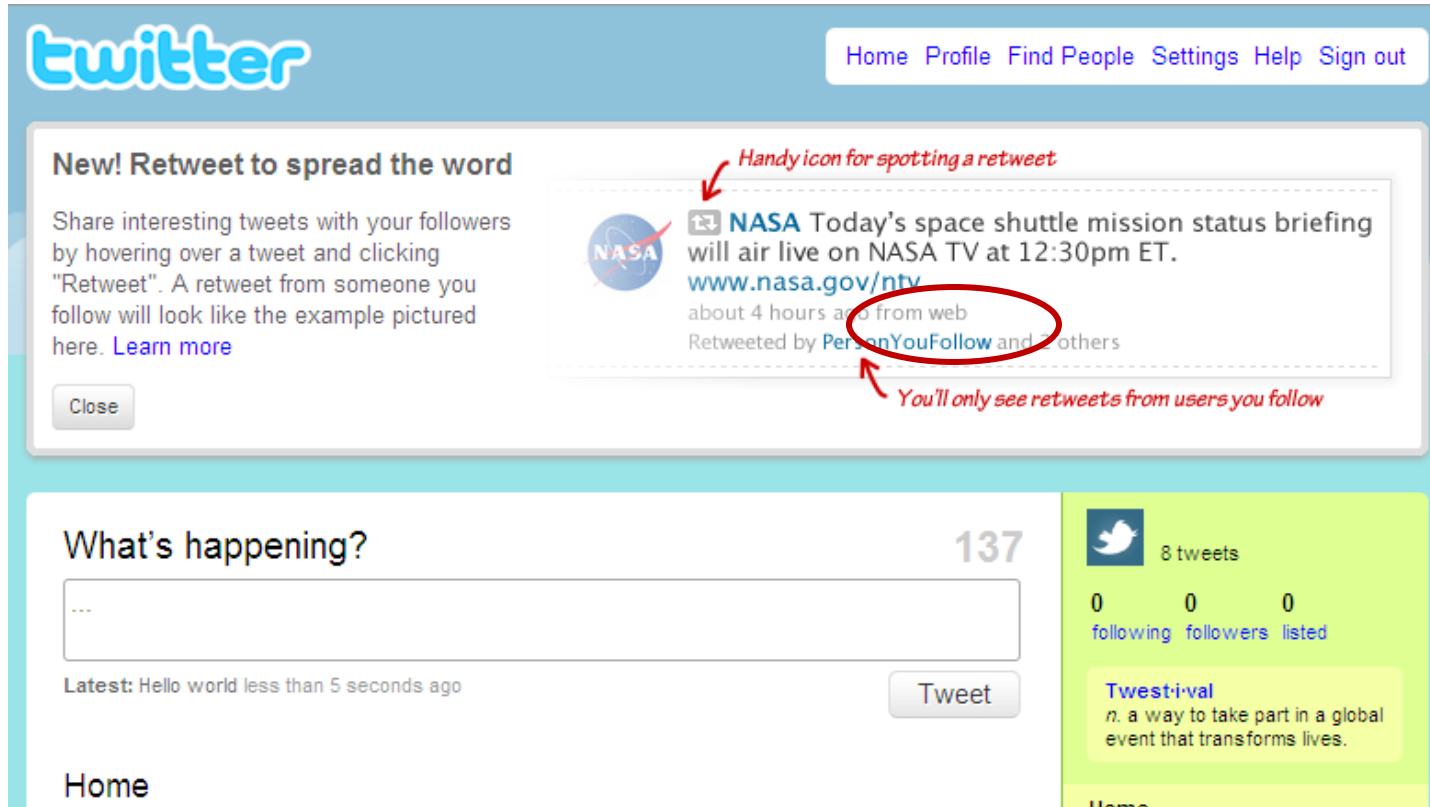
or, the problem with pixels

The Problem with Pixels



When the attack is prepared...

The Problem with Pixels



When the attack is executed... Oops!

- Remember this - ``
`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">`
- `http://en.wikipedia.org/w/index.php?title=Clickjacking&action=edit#wpSave`

First name:	<input type="text" value="Alice"/>
Last name:	<input type="text" value="Foo"/>
Nickname:	<input type="text"/>
Zip code: <i>(optional)</i>	<input type="text"/>
Country: <i>(optional)</i>	<input type="text" value="loc"/>
Time zone:	<input type="text" value="(GMT+00:00) London"/>
	<input checked="" type="checkbox"/> Display all timezones

nonGmailAlternates

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.

(Primary email)

Add an additional email address:

Element IDs on Google Accounts page



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

```
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

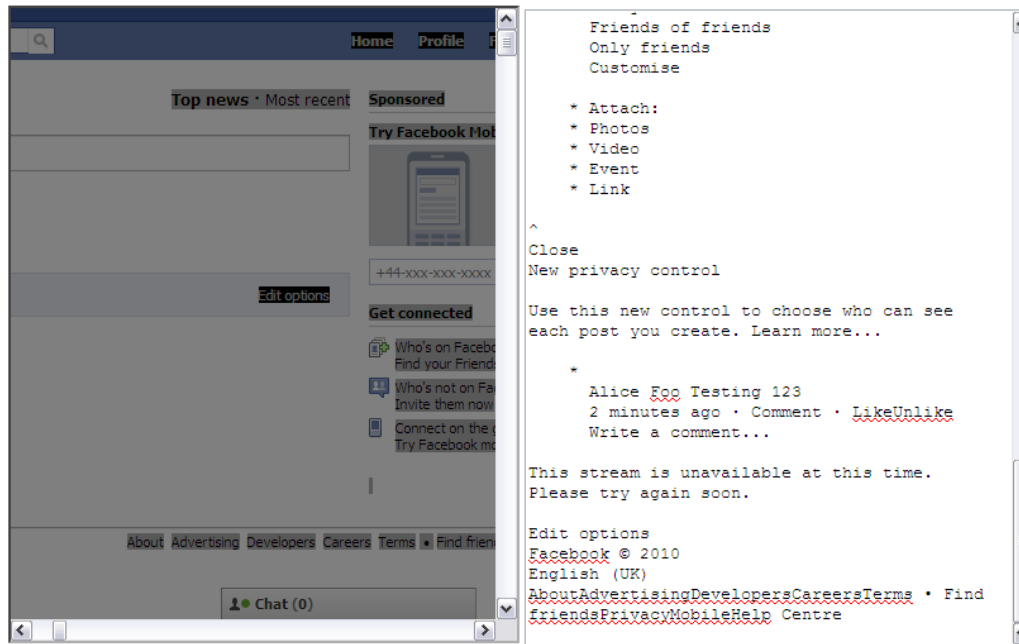
- Reverse drag and drop
 - Drag content from inside iframe
- ```
<body ondrop="alert(event.dataTransfer.getData('Text'))">
```
- Links and images are draggable as URLs

- Links may contain sensitive information
  - Hashes, Object IDs, User information

[https://docs.google.com/Doc?docid=0Acwo2Bn17-PrZGZudHRobnJfNDNmOTZzOTIkbg&hl=en\\_GB](https://docs.google.com/Doc?docid=0Acwo2Bn17-PrZGZudHRobnJfNDNmOTZzOTIkbg&hl=en_GB)

- ...but we can do better

- 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...



# Content Extraction – Steps

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

## Top Secret

A

Nunc nec arcu tortor. Sed laoreet turpis non libero consectetur id lacinia est cursus. Nunc arcu nulla, iaculis vel luctus ut, consequat nec justo. Nullam fringilla dignissim elit, nec ornare nibh iaculis ac. Donec erat ligula, pulvinar id facilisis in, pulvinar dignissim erat. Proin hendrerit, **A** non dignissim adipiscing, libero libero vulputate urna, quis lacinia magna quam eget lorem. Duis blandit est id lectus ultricies, **C** tempus neque feugiat. Curabitur ac libero eros. Nullam euismod convallis tortor ac interdum. Nulla varius, nulla varius volutpat, **C** tempor, ipsum nunc fringilla eros, in placerat ante leo vel urna. Ut vulputate, lorem in condimentum gravida, leo lectus semper enim, ac pretium mauris arcu vel orci. Proin vehicula erat vel arcu dapibus ac scelerisque magna gravida. In suscipit tristique nunc, scelerisque iaculis urna convallis sit amet. Praesent lobortis viverra nibh, sed pellentesque nisi tristique eu. Phasellus adipiscing malesuada elit. Duis quis vehicula massa. Sed vel tempor mi. Donec tellus lorem, bibendum nec imperdiet sed, egestas vitae ligula. Phasellus nec velit sem, sit amet iaculis est.

Donec lacinia auctor nunc vitae laoreet. Maecenas ligula elit, facilisis eget convallis eget, commodo id dolor. Fusce gravida feugiat turpis, eget elementum purus feugiat vel. Etiam vitae lorem non mi faucibus sollicitudin. Integer tempor dapibus rhoncus. Maecenas pretium ultricies porttitor. Etiam rutrum risus sed ipsum mollis auctor. Sed varius augue et nibh consectetur fringilla. Praesent a elit est. In lobortis nisi id metus interdum et fermentum nisi aliquam. Nunc in velit diam. Donec mattis libero sit amet mi facilisis ac cursus est consectetur. Pellentesque pulvinar enim at diam tristique viverra. Donec dignissim egestas lacus vel mollis. Proin lobortis pellentesque mauris a pulvinar. Suspendisse sed turpis sapien. Vivamus iaculis, dui vel gravida faucibus, mauris nisi suscipit turpis, quis gravida nisi nisi ut libero. Aenean et lorem id augue ultricies faucibus. Donec vehicula, mauris id ornare sagittis, est sem molestie enim, non pellentesque metus est ac tellus. Curabitur ipsum lorem, ullamcorper in fringilla vel, tempus eget enim.

Proin porttitor luctus mi ac faucibus. Proin eget nibh rutrum nunc eleifend sollicitudin. Nulla placerat, sapien vel pellentesque mollis, enim purus pharetra nisi, elementum euismod metus risus cursus velit. Phasellus dui turpis, tempus et pharetra nec, sollicitudin a nisi. Suspendisse in aliquam metus. Proin tristique ullamcorper ultricies. Praesent porta pretium tortor, ut consectetur elit lobortis nec. Donec at metus libero, id porta nibh. Aliquam tristique justo quis purus luctus eu auctor magna dignissim. Vivamus est lorem, vehicula in tincidunt ut, pharetra at diam. Ut scelerisque, tortor nec varius vulputate, urna quam porta orci, eget sagittis ante est eu tortor. Ut hendrerit tempor enim ut aliquet. In dignissim bibendum fermentum. Donec consequat est ut turpis varius sit amet egestas sem gravida. Quisque ipsum mi, molestie sit amet rutrum eget, vulputate euismod nisi. Suspendisse pretium, massa vel sagittis fermentum, sem nisi adipiscing velit, egestas malesuada mi nisi vitae nibh. Nulla faucibus viverra dolor quis congue. Morbi ornare mauris quis est posuere at venenatis est sodales.

Sed adipiscing nisi ac nisi sagittis accumsan. Aenean volutpat, ante ac mollis scelerisque, urna mauris varius neque, nec feugiat arcu orci sit amet magna. Nullam est ipsum, volutpat eget varius id, accumsan ut augue. Praesent facilisis lacus tempor erat vehicula laoreet. Phasellus lobortis, risus non volutpat tincidunt, urna augue dignissim tortor, sed vehicula purus purus vel tortor. Vestibulum nec urna quis est elementum lobortis eu nec quam. Nunc et enim et mauris blandit fringilla quis a nisi. Phasellus vitae nunc mauris, id sollicitudin urna. Sed id dui sapien, in rhoncus mi. In tempor, risus eu commodo gravida, purus sapien aliquam diam, non tempor odio eros a turpis.

Nulla vehicula turpis ac ipsum aliquam vulputate. Vivamus ac aliquam neque. Praesent magna enim, condimentum vitae dapibus sed, pharetra ac erat. Vestibulum sit amet velit felis, sed posuere est. Morbi purus felis, tincidunt ut ornare nec, consectetur a nisi. Nulla arcu metus, bibendum nec lobortis eget, gravida ut ipsum. Mauris in faucibus leo. Fusce porta ipsum ac mi semper iaculis. Duis tristique suscipit tortor dapibus hendrerit. Quisque a eros eu felis congue laoreet. Donec tempus justo neque. Curabitur felis enim, tincidunt eget fermentum non, consequat ut tortor. Nullam lorem odio, placerat id egestas eget, tempus in sem. Sed lorem est, varius nec iaculis a, porta in risus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Integer placerat iaculis nulla, quis accumsan leo feugiat ut. Praesent fermentum mollis sem, sed semper ipsum volutpat eu. Etiam hendrerit scelerisque orci, vitae elementum eros convallis vitae.

B



- 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





- 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

- 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

- 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

- 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

- 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

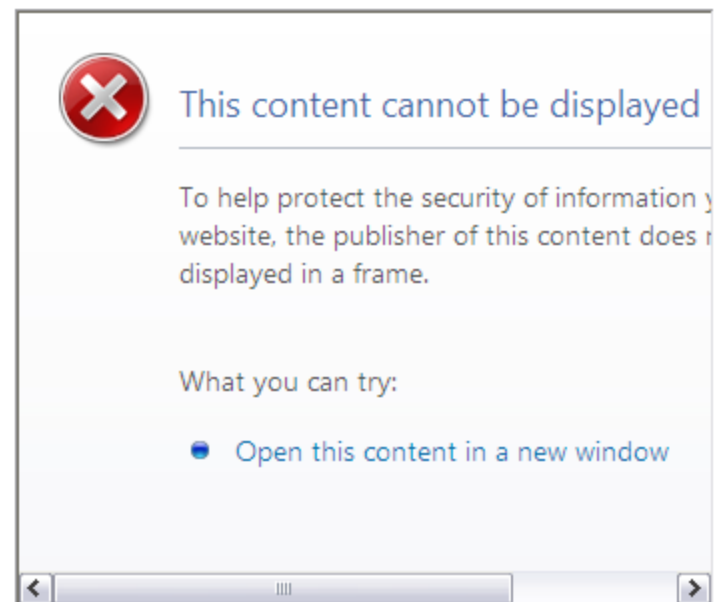
- More targeted attacks
- Check for login page IDs
  - Is a user logged into site X
  - Check if 're-authentication' is needed for sensitive pages
- Check position of page footer
  - How many emails in your inbox
  - How many results for search query X
- Brute force numeric IDs
  - What items in your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



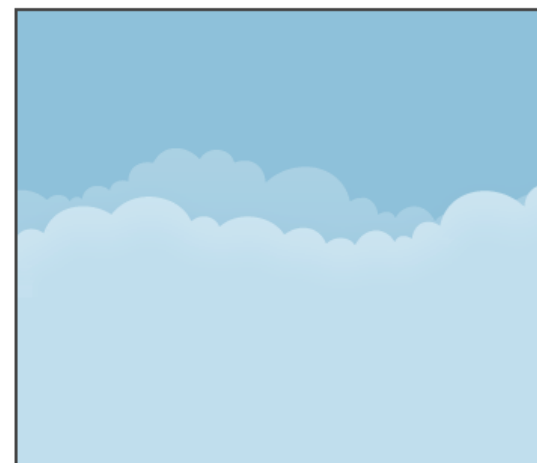
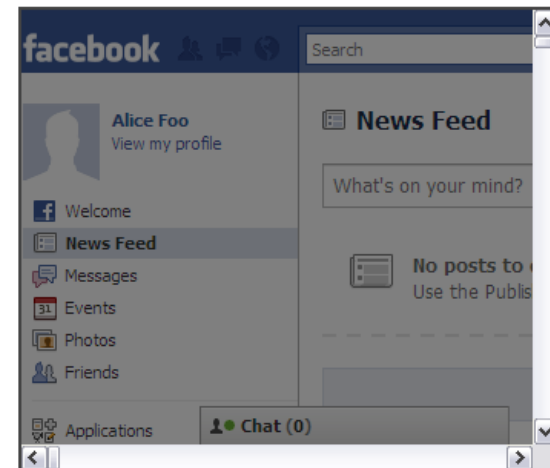
# Clickjacking Countermeasures

and how to break them

- 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



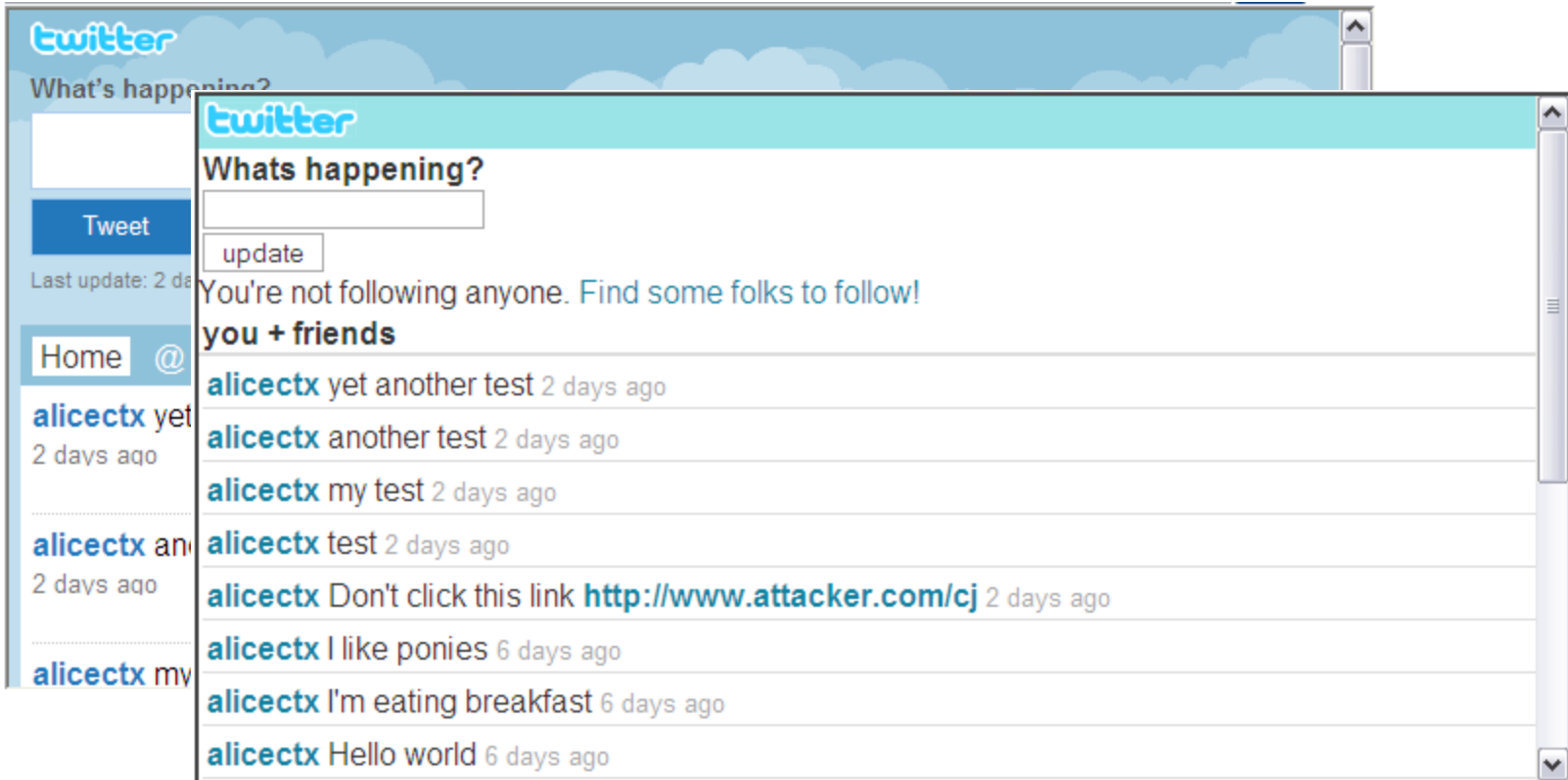
- JavaScript protection
  - Framebusting
  - Hide or obscure content
- Facebook and Twitter use JavaScript protection but not X-Frame-Options
- JavaScript protection is not 100% effective
- Even if it was, most sites still have unprotected areas



- 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

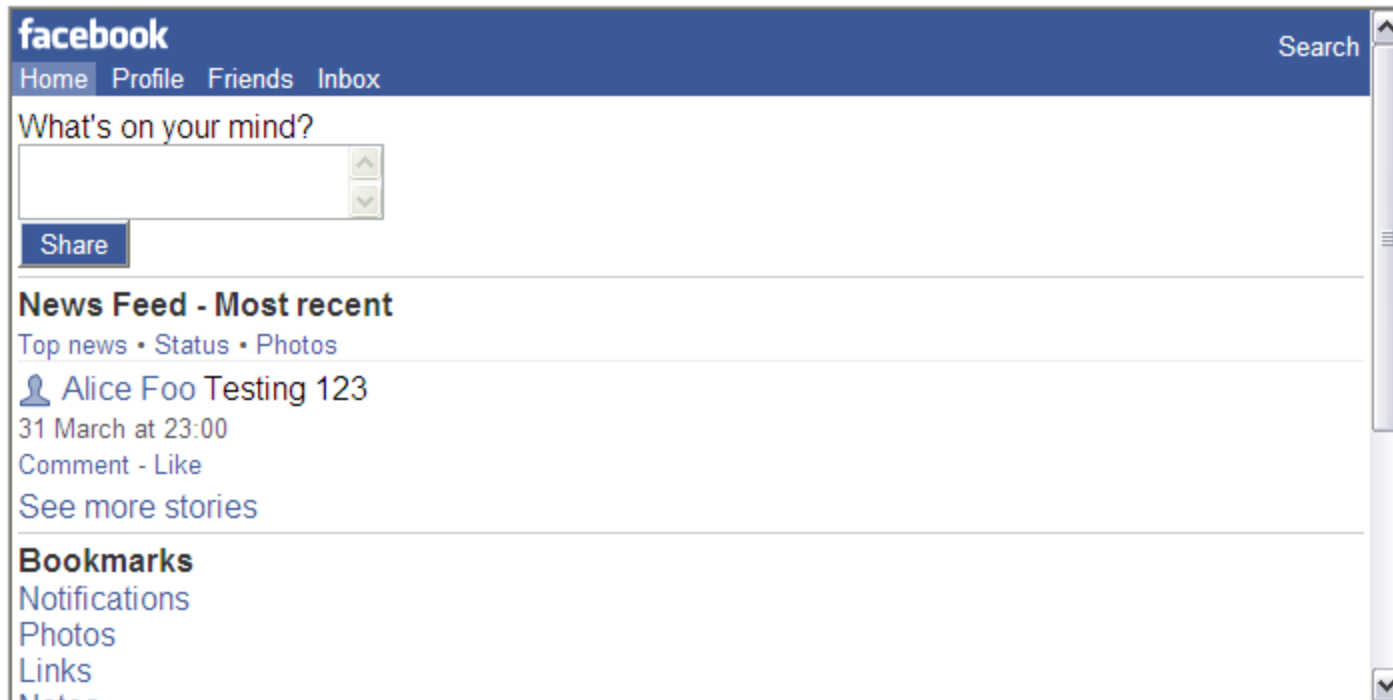


- High profile sites have implemented Clickjacking protection
- Most are still vulnerable through
- Mobile sites
- Gadgets / widgets allow framing

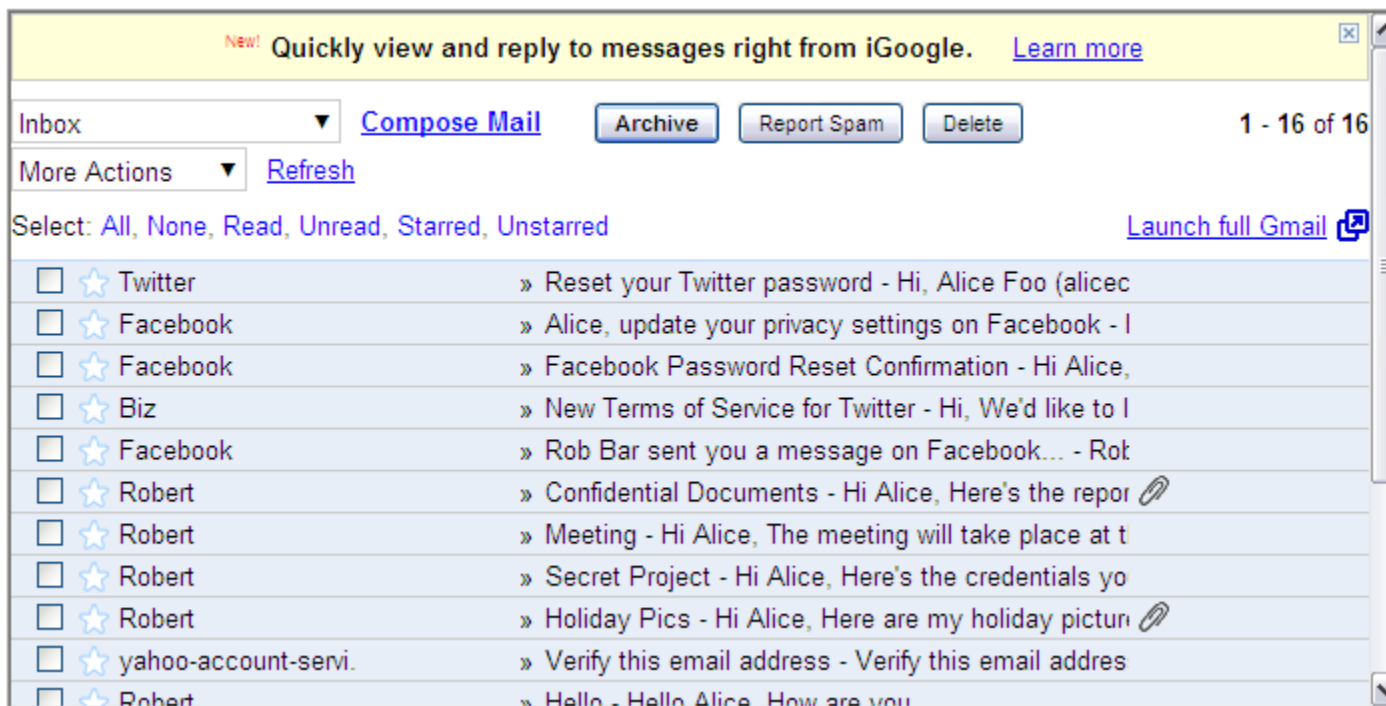


<http://mobile.twitter.com>

<http://m.twitter.com>




<http://m.facebook.com>





New! Quickly view and reply to messages right from iGoogle. [Learn more](#)

Inbox ▼ [Compose Mail](#) [Archive](#) [Report Spam](#) [Delete](#) 1 - 16 of 16

More Actions ▼ [Refresh](#)

Select: [All](#), [None](#), [Read](#), [Unread](#), [Starred](#), [Unstarred](#) [Launch full Gmail](#) 

<input type="checkbox"/>	★ Twitter	» Reset your Twitter password - Hi, Alice Foo (alicec
<input type="checkbox"/>	★ Facebook	» Alice, update your privacy settings on Facebook - I
<input type="checkbox"/>	★ Facebook	» Facebook Password Reset Confirmation - Hi Alice,
<input type="checkbox"/>	★ Biz	» New Terms of Service for Twitter - Hi, We'd like to I
<input type="checkbox"/>	★ Facebook	» Rob Bar sent you a message on Facebook... - Rot
<input type="checkbox"/>	★ Robert	» Confidential Documents - Hi Alice, Here's the repor 
<input type="checkbox"/>	★ Robert	» Meeting - Hi Alice, The meeting will take place at t
<input type="checkbox"/>	★ Robert	» Secret Project - Hi Alice, Here's the credentials yo
<input type="checkbox"/>	★ Robert	» Holiday Pics - Hi Alice, Here are my holiday pictur 
<input type="checkbox"/>	★ yahoo-account-servi.	» Verify this email address - Verify this email addres
<input type="checkbox"/>	★ Robert	» Hello - Hello Alice, How are you

<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

# Questions