Clickjacking

Vitaly Shmatikov
Reading Assignment

◆ “Next Generation Clickjacking”
◆ “Clickjacking: Attacks and Defenses”
Clickjacking (UI Redressing)

- Attacker overlays multiple transparent or opaque frames to trick a user into clicking on a button or link on another page

- Clicks meant for the visible page are hijacked and routed to another, invisible page
Clickjacking in the Wild

- Google search for “clickjacking” returns 624,000 results... this is not a hypothetical threat!
- Summer 2010: Facebook worm superimposes an invisible iframe over the entire page that links back to the victim's Facebook page
  - If victim is logged in, automatically recommends link to new friends as soon as the page is clicked on
- Many clickjacking attacks against Twitter
  - Users send out tweets against their will
Clickjacking Meets Spamming

Facebook sues alleged clickjacking spammer sparking row

Facebook is suing a marketing firm, accusing it of "spreading spam through misleading and deceptive tactics".

Adscend Media is alleged to have carried out "clickjacking".

The practice involves placing posts on the social network which include code that causes the links to appear on the users' homepages as
It’s All About iFrame

◆ Any site can frame any other site

```html
<iframe src="http://www.google.com/...">
</iframe>
```

◆ HTML attributes

- **Style**
- **Opacity** defines visibility percentage of the iframe
  - 1.0: completely visible
  - 0.0: completely invisible
Hiding the Target Element

- Use CSS `opacity` property and `z-index` property to hide target element and make other element float under the target element.
- Using CSS `pointer-events: none` property to cover other element over the target element.

[“Clickjacking: Attacks and Defenses”]
Partial Overlays and Cropping

- Overlay other elements onto an iframe using CSS \texttt{z-index} property or Flash Window Mode \texttt{wmode=direct} property.
- Wrap target element in a new iframe and choose CSS position offset properties.

[“Clickjacking: Attacks and Defenses”]
Drag-and-Drop API

- Modern browsers support drag-and-drop API
- JavaScript can use it to set data being dragged and read it when it’s dropped
- Not restricted by the same origin policy: data from one origin can be dragged to a frame of another origin
  - Reason: drag-and-drop can only be initiated by user’s mouse gesture, not by JavaScript on its own
Abusing Drag-and-Drop API

[“Next Generation Clickjacking”]

1. Bait the user to click and start dragging
2. Invisible iframe with attacker’s text field under mouse cursor, use API to set data being dragged
3. Invisible iframe from another origin with a form field

With two drag-and-drops (simulated scrollbar, etc.), can select and extract arbitrary content from another origin

Frog. Blender. You know what to do.
Fake Cursors

- Use CSS `cursor` property and JavaScript to simulate a fake cursor icon on the screen

Real cursor icon

Fake cursor icon

cursor: none
Keyboard “Strokejacking”

Simulate an input field getting focus, but actually the keyboard focus is on target element, forcing user to type some unwanted information into target element.

Attacker’s page

Typing Game
Type whatever screen shows to you

Xfpg95403poigr06=2kfpx

Hidden iframe within attacker’s page

Bank Transfer
Bank Account: 9540
Amount: 3062 USD

Transfer
Compromising Temporal Integrity

- Manipulate UI elements after the user has decided to click, but before the actual click occurs

[“Clickjacking: Attacks and Defenses”]
Cursor Spoofing

[“Clickjacking: Attacks and Defenses”]
Double-Click Attack

[“Clickjacking: Attacks and Defenses”]

◆ Bait the user to perform a double-click, switch focus to a popup window under the cursor right between the two clicks
Whack-A-Mole Attack

Ask the user to click as fast as possible, suddenly switch Facebook Like button

Instructions:
Please click on blue buttons as fast as possible. The faster you complete this game, the greater your chances to win a $100 prize! If you don't click on a button, the game will skip it in 10 seconds.

Buttons clicked: 17/20
Time elapsed: 27.6 sec
Solution: Frame Busting

◆ I am a page owner
◆ All I need to do is make sure that my web page is not loaded in an enclosing frame ...

Clickjacking: solved!
  • Does not work for FB “Like” buttons and such, but Ok
◆ How hard can this be?

```javascript
if (top != self)
    top.location.href = location.href
```
Frame Busting in the Wild

Survey by Gustav Rydstedt, Elie Burzstein, Dan Boneh, Collin Jackson

Following slides shamelessly jacked from Rydstedt
### Conditional Statements

<table>
<thead>
<tr>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>if (top != self)</td>
</tr>
<tr>
<td>if (top.location != self.location)</td>
</tr>
<tr>
<td>if (top.location != location)</td>
</tr>
<tr>
<td>if (parent.frames.length &gt; 0)</td>
</tr>
<tr>
<td>if (window != top)</td>
</tr>
<tr>
<td>if (window.top !== window.self)</td>
</tr>
<tr>
<td>if (window.self != window.top)</td>
</tr>
<tr>
<td>if (parent &amp;&amp; parent != window)</td>
</tr>
<tr>
<td>if (parent &amp;&amp; parent.frames &amp;&amp; parent.frames.length &gt; 0)</td>
</tr>
<tr>
<td>if((self.parent&amp;&amp; !(self.parent===self))&amp;&amp; (self.parent.frames.length!=</td>
</tr>
</tbody>
</table>

If My Frame Is Not On Top ...
<table>
<thead>
<tr>
<th>Counter-Action Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>top.location = self.location</code></td>
</tr>
<tr>
<td><code>top.location.href = document.location.href</code></td>
</tr>
<tr>
<td><code>top.location.href = self.location.href</code></td>
</tr>
<tr>
<td><code>top.location.replace(self.location)</code></td>
</tr>
<tr>
<td><code>top.location.href = window.location.href</code></td>
</tr>
<tr>
<td><code>top.location.replace(document.location)</code></td>
</tr>
<tr>
<td><code>top.location.href = window.location.href</code></td>
</tr>
<tr>
<td><code>top.location.href = &quot;URL&quot;</code></td>
</tr>
<tr>
<td><code>document.write('')</code></td>
</tr>
<tr>
<td><code>top.location = location</code></td>
</tr>
<tr>
<td><code>top.location.replace(document.location)</code></td>
</tr>
<tr>
<td><code>top.location.replace('URL')</code></td>
</tr>
<tr>
<td><code>top.location.href = document.location</code></td>
</tr>
<tr>
<td><code>top.location.replace(window.location.href)</code></td>
</tr>
<tr>
<td><code>top.location.href = location.href</code></td>
</tr>
<tr>
<td><code>self.parent.location = document.location</code></td>
</tr>
<tr>
<td><code>parent.location.href = self.document.location</code></td>
</tr>
<tr>
<td><code>top.location.href = self.location</code></td>
</tr>
<tr>
<td><code>top.location = window.location</code></td>
</tr>
<tr>
<td><code>top.location.replace(window.location.pathname)</code></td>
</tr>
</tbody>
</table>
What About My Own iFrames?

- Check: *is the enclosing frame one of my own?*
- How hard can this be?
- Survey of several hundred top websites ...
  ... *all frame busting code is broken!*
if (top.location != location) {
    if (document.referer &&
        document.referer.indexOf("walmart.com") == -1)
    {
        top.location.replace(document.location.href);
    }
}
Error in Referer Checking

<iframe src="http://www.walmart.com">
if (window.self != window.top &&
    !document.referer.match(/https?://[^?/]+.nytimes\..com\//)) {
    self.location = top.location;
}
Error in Referer Checking

<iframe src="http://www.nytimes.com">
if (self != top) {
    var domain = getDomain(document.referer);
    var okDomains = /usbank|localhost|usbnet/;
    var matchDomain = domain.search(okDomains);

    if (matchDomain == -1) {
        // frame bust
    }
}
Error in Referer Checking

From http://usbank.attacker.com/
<iframe src="http://www.usbank.com">
Strategic Relationship?

Norwegian State House Bank
http://www.husbanken.no
Strategic Relationship?

Bank of Moscow

http://www.rusbank.org
try{
    A=!top.location.href
} catch(B){}
A=A&&
!(document.referer.match(/^https?:\/[\/-az09.]*\.(google\.(co\.|com\.)? [a-z] +/imgres/i))&&
!(document.referer.match(/^https?:\/%([^\%/]*\.)? (myspace\.|com|myspace\.|cn|simsidekick\.|com|levisawards\.|com|digg\.|com)\//i));

if(A){ // Frame bust }
Do Your Trusted Sites Frame Bust?

Google Images does not frame bust
Many Attacks on Referer Header

- Open redirect referer changer
- HTTPS -> HTTP redirect changes the header
- Apparently, hard to get regular expression right
- Trust other sites to frame your pages, but what if those trusted sites can be framed themselves?
if(top.location != self.location) {
    parent.location = self.location;
}
Who Is Your Daddy Parent?

Double framing!!

framed1.html
<iframe src="framed2.html" >

framed2.html
<iframe src="victim.com" >
Who Is On Top?

```python
if (top.location != self.location):
    top.location = self.location
```

If `top.location` can be changed or disabled, this code is useless.
Location Clobbering

◆ IE 7

```javascript
var location="clobbered";
```

◆ Safari

```javascript
window.__defineSetter__("location", function(){
  top.location now undefined
});
```
User Can Stop Frame Busting

- User can manually cancel any redirection attempt made by frame busting code
- Attacker just needs to ask...

```html
<script>
    window.onbeforeunload = function() {
        return "Do you want to leave PayPal?";
    }
</script>
<iframe src="http://www.paypal.com">
```
Ask Nicely
... Or Don’t Even Ask

Most browsers let **attacker** cancel the relocation programmatically

var prevent_bust = 0
window.onbeforeunload = function() {kill_bust++ }
setInterval(function() {
    if (kill_bust > 0) {
        kill_bust -= 2;
        window.top.location = 'http://no-content-204.com'
    }
}, 1);
<iframe src="http://www.victim.com">
X-Frame-Options

- HTTP header sent with the page
- Two possible values: DENY and SAMEORIGIN
- DENY: page will not render if framed
- SAMEORIGIN: page will only render if top frame has the same origin
Adoption of X-Frame-Options

- Good adoption by browsers
- Poor adoption by sites
- Limitations
  - Per-page policy
  - No whitelisting of origins
  - Proxy problems
Content Security Policy (Firefox 4)

- Another HTTP header: `frame-ancestors` directive can specify allowed framers
- Allows specific restrictions and abilities per site
Best For Now (Still Not Good)

<style>html { visibility: hidden } </style>
<script>
if (self == top) {
    document.documentElement.style.visibility = 'visible';
} else {
    top.location = self.location;
}
</script>
These Sites Do Frame Busting
Do These?
## Frame Busting on Mobile Sites

<table>
<thead>
<tr>
<th>Site</th>
<th>URL</th>
<th>Framebusting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td><a href="http://m.facebook.com/">http://m.facebook.com/</a></td>
<td>YES</td>
</tr>
<tr>
<td>MSN</td>
<td><a href="http://home.mobile.msn.com/">http://home.mobile.msn.com/</a></td>
<td>NO</td>
</tr>
<tr>
<td>GMail</td>
<td><a href="http://m.gmail.com">http://m.gmail.com</a></td>
<td>NO</td>
</tr>
<tr>
<td>Baidu</td>
<td><a href="http://m.baidu.com">http://m.baidu.com</a></td>
<td>NO</td>
</tr>
<tr>
<td>Twitter</td>
<td><a href="http://mobile.twitter.com">http://mobile.twitter.com</a></td>
<td>NO</td>
</tr>
<tr>
<td>MegaVideo</td>
<td><a href="http://mobile.megavideo.com/">http://mobile.megavideo.com/</a></td>
<td>NO</td>
</tr>
<tr>
<td>Tube8</td>
<td><a href="http://m.tube8.com">http://m.tube8.com</a></td>
<td>NO</td>
</tr>
<tr>
<td>PayPal</td>
<td><a href="http://mobile.paypal.com">http://mobile.paypal.com</a></td>
<td>NO</td>
</tr>
<tr>
<td>USBank</td>
<td><a href="http://mobile.usbank.com">http://mobile.usbank.com</a></td>
<td>NO</td>
</tr>
<tr>
<td>First Interstate Bank</td>
<td><a href="http://firstinterstate.mobi">http://firstinterstate.mobi</a></td>
<td>NO</td>
</tr>
<tr>
<td>NewEgg</td>
<td><a href="http://m.newegg.com/">http://m.newegg.com/</a></td>
<td>NO</td>
</tr>
<tr>
<td>MetaCafe</td>
<td><a href="http://m.metacafe.com/">http://m.metacafe.com/</a></td>
<td>NO</td>
</tr>
<tr>
<td>RenRen</td>
<td><a href="http://m.renren.com/">http://m.renren.com/</a></td>
<td>NO</td>
</tr>
<tr>
<td>MySpace</td>
<td><a href="http://m.myspace.com">http://m.myspace.com</a></td>
<td>NO</td>
</tr>
<tr>
<td>VKontakte</td>
<td><a href="http://pda.vkontakte.ru/">http://pda.vkontakte.ru/</a></td>
<td>NO</td>
</tr>
<tr>
<td>WellsFargo</td>
<td><a href="https://m.wf.com/">https://m.wf.com/</a></td>
<td>NO</td>
</tr>
<tr>
<td>NyTimes</td>
<td><a href="http://m.nytimes.com">http://m.nytimes.com</a></td>
<td>Redirect</td>
</tr>
<tr>
<td>E-Zine Articles</td>
<td><a href="http://m.ezinearticles.com">http://m.ezinearticles.com</a></td>
<td>Redirect</td>
</tr>
</tbody>
</table>
Tapjacking

- Zoom buttons in a transparent iframe so that they cover entire screen
- Hide or fake URL bar
- Make a page that masquerades as a known application to trick user into clicking

Read more:
http://seclab.stanford.edu/websec/framebusting/