

Clickjacking (UI Redress)

- When the attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page
- Clicks are “hijacked” and routed to another page
- You can kind of think of it as a more involved and more “physical” CSRF attack

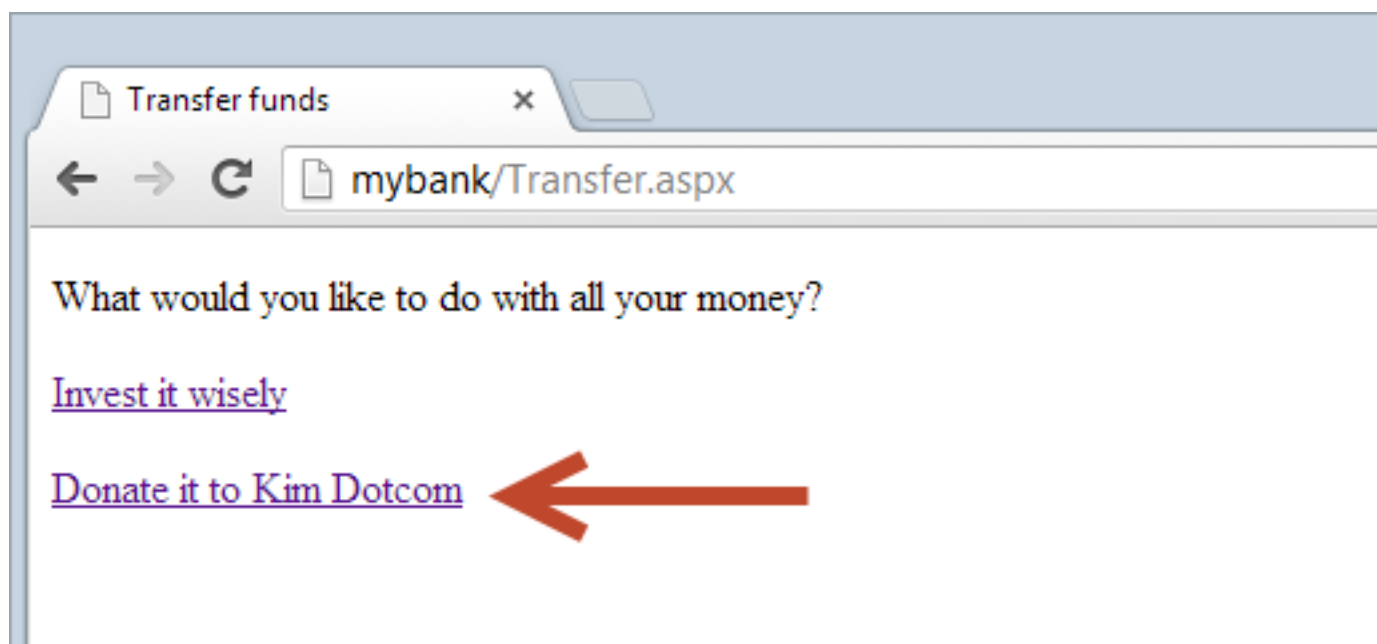
Example

- Assumption: victim is logged into social network
- Imagine a page with a “click to view embarrassing photos” link
- On top of that page (and invisible to the user) the attack has loaded an iFrame with their page on a social network
- The attacker lines up the “add friend” button on their profile page up with the “click to view” link
- When the user clicks to view the embarrassing photos they are actually clicking the “add friend” button
- The user is tricked into adding the attacker as their friend on the social network



Figure 1: Visualization of a clickjacking attack on Twitter's account deletion page.







Code Example

```
<div style="position: absolute; left: 10px; top: 10px;">Hey - we're giving away iPad minis!!! Just click  
the WIN button and it's yours!!!</div>  
<div style="position: absolute; left: 200px; top: 50px;">  
    
</div>  
<div style="position: absolute; left: 10px; top: 101px; color: red; font-weight: bold;">>> WIN <<</div>  
<iframe style="opacity: 0;" height="545" width="680" scrolling="no" src="http://mybank/Transfer.aspx">  
</iframe>
```


Two Main Defenses

- X-Frame-Options HTTP response header
- Framebusting UI code

X-Frame-Options header

- The only “real” solution
- Tells the browser whether a requested page can be framed
- Can be tweaked per page
- Three values:
 - DENY - no framing period
 - SAMEORIGIN - can only be framed in the same domain
 - ALLOW-FROM - can only be framed by trusted pages

Framebusting UI code

- Check to see if the page is same as the one loaded from the address bar

```
if (top.location != location) {  
    top.location = self.location;  
}
```

Common Defenses

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 > 0)</code>
5.5%	<code>if (window != top)</code>
5.5%	<code>if (window.top !== window.self)</code>
2%	<code>if (window.self != window.top)</code>
2%	<code>if (parent && parent != window)</code>
2%	<code>if (parent && parent.frames && parent.frames.length>0)</code>
2%	<code>if((self.parent&&!(self.parent===self))&&(self.parent.frames.length!=0))</code>

Table 2: Frame busting conditional statement

Can anyone think of bypasses to these?

Framebusting Arms Race

- Frame buster busters such as:
 - Nesting the victim site in two frames as the double framing causes the descendent frame navigation policy to disable redirection
 - Tapping into the `onBeforeUnload` event to cancel the redirection (albeit with some user input) when the frame buster attempts to unload the page
 - Exploiting XSS filters designed to prohibit Cross-Site Scripting in order to cancel out the frame buster

Can you think of a bypass?

USBank framebusting

```
if (self != top) {  
  var dom = getDom(document.referrer);  
  var okDom = /usbank|localhost|usbnet/;  
  var matchDomain = dom.search(okDom);  
  
  if (matchDomain == -1) { //bust }
```


USBank framebusting issues

- Still allows domains with the word usbank in it
- Can be framed by:
 - <http://www.husbanken.no>
 - <http://www.rusbank.org>
 - Or any domain registered by an attacker with usbank in it

Myspace framebusting

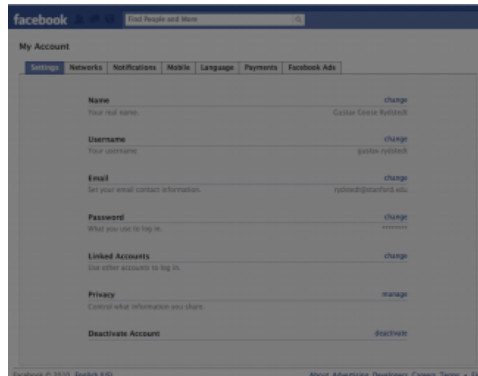
```
try{
  A=!top.location.href
}catch(B){}
A=A&&!(document.referrer.match(
  /^https?:\/\/\[ -a-z0-9.\]*\.google\.(co\.|
  |com\.)?[a-z]+\imgres/i))
&&!(document.referrer.match(
  /^https?:\/\/\[^\]*\.(myspace\.com
  |myspace\.cn
  |simsidekick\.com
  |levisawards\.com\/\/i));
if(A) { // frame bust }
```

Myspace framebusting issues

- Allows for framing by Google images
- Google images does not employ frame busting
- An attack simply frames Google images and then cause Google images to frame Myspace

A Clever Clickjacking Protection

- Facebook.com inserts a gray semi-transparent div that covers all of the content when a profile page is framed
- Allows framing but blocks clickjacking attacks



```
if (top !== self) {  
  window.document.write('<div style=  
    'background: black; opacity: 0.5;  
    filter: alpha(opacity = 50);  
    position: absolute; top: 0px; left: 0px;  
    width: 9999px; height: 9999px;  
    z-index: 1000001'  
    onClick='top.location.href=window.location.href'  
></div>');  
}
```

Facebook.com bypass

- Make the enclosing frame so large that the center of the frame is outside of the dark div
- The content centers itself automatically
- The scrollTo function dynamically scrolls to the center

```
<body style="overflow-x:hidden;
border:0px;margin:0px;">

<iframe width="21800px" height="2500px"
src="http://facebook.com/"
frameborder="0"
marginheight="0" marginwidth="0" >
</iframe>

<script> window.scrollTo(10200,0);
</script>
```

Resources

- <http://www.troyhunt.com/2013/05/clickjack-attack-hidden-threat-right-in.html>
- <http://seclab.stanford.edu/websec/framebusting/framebust.pdf>
- <https://www.owasp.org/index.php/Clickjacking>