Bamboozling *Jon Doe* Part 2

For this part of the assignment, I used an `iframe` in an HTML page. An `iframe` is used to display a document within a document. More specifically, you can use an `iframe` to embed another HTML page within your current page. Using `iframes` we can carry out a click jacking attack.

The Attack

My custom HTML page uses an `iframe` to include my profile page (*oneNutWonder :: id=36*) from armbook. I then use CSS to move the embedded page around to put the `Add Friend` button directly on top of a button that says `YOU WON!`. I set the opacity of the `iframe` to zero so that the embedded page is invisible to visitors. Now when they click the button that says `YOU WON!` while they are logged in to armbook in a different tab, they will friend my profile. All of this hapens without the user knowing what they just did.

Mitigating Clickjacking

One method of preventing this attack is using CSP frame-ancestor directives. These directives allow a site to say whether or not their content is allowed to be rendered in a `<frame>` or `<iframe>` tag, or disallow rendering all together. In addition to CSP frame-ancestor directives, sites can use X-Frame-Options headers. If a site uses `DENY` for this header, it will prevent any domain from embedding its content.

Resource: `https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html`