All this text wasn’t written by me (Tiago a.k.a TigaxMT).

All of it was transcribed by me from HackerOne video lessons.

All credits and thanks go to them.

Clickjacking

Probably the most common form of this attack is “likejacking”. This generally takes the form of an ad-covered page with what looks like a video player in the middle.
When the user clicks the play button on the player, nothing happens. That’s because they really just clicked a Facebook ‘Like’ button, posting the video site to their wall and spreading this to their friends, all without their knowledge.

HOW IT WORKS

The page you want to surreptitiously control is placed into a Iframe, laid over the page you control, made transparent by CSS.
By enticing a user to click an element on the page you control, you’re really causing them to click on an element of the victim page.

VARIATIONS

Obviously this can be used for many, situations, not just likejacking.
One of the most interesting clickjacking techniques involved duplicating your mouse cursor, a fixed distance away from the original, then hiding the original. The victim then would click on safe elements with the new cursor, when in reality they’re clicking on unsafe elements with the original cursor.

FALSE REPORTING

While clickjacking does appear in many applications in various forms, it’s often reported when there is no actual security impact. Unless the user can perform authorized/sensitive actions on a page, clickjacking is just an oversight and not security issue.

MITIGATION

- Framekiller JS sent to clients will break out of the Iframe, mitigating clickjacking
  - But IE has serious issues here
- Content Security Policy headers can mitigate this – we’ll talk more about those later
- The X-Frame-Options header can restrict which origins can embed a given page, mitigating clickjacking.

By far, the most effective mitigation is the X-Frame-Options header. This is supported in every major browser and allows for complete prevention of clickjacking attacks.
Simply setting this to DENY or SAMEORIGIN will prevent an attacker-controlled site from embedding your page and executing a clickjacking attack.