Clickjacking Attacks

W5.1. Please describe two common approaches used by clickjacking attacks?

W5.2. Please answer these questions based on the following HTML page:

1. Which iframe is on the top and which one is on the bottom?
2. If we click on this page, where does the click go?
3. In a typical clickjacking attack, which page (the target page or the attacker's page) is placed on the top?
4. Please decide the opacity value for X and Y for a typical clickjacking attack.

```
<iframe id="A" src="http://www.example99.com/"
        style="border:0px; width:800px; height:1500px;">
</iframe>
<iframe id="B" src="http://www.example32.com/"
        style="border:0px; width:800px; height:1500px;">
</iframe>
<style type="text/css">
   #A {position:absolute; top:0px; left:0px; opacity: X}
   #B {position:absolute; top:0px; left:0px; opacity: Y}
</style>
```

W5.3. Please construct two iframes, such that one of the iframes seems to be part of the page in another iframe. Please then describe how this setup can be used in clickjacking attacks.

W5.4. The followings are the responses from a web server `www.example32.com`. Each of these responses is placed inside an iframe. (1) If the host page of these iframes come from `www.example32.com`, which of the following pages can be displayed? (2) If the host page come from `www.example99.com`, which of the following pages can be displayed?

```
Page 1:
<?php
  $csp= "Content-Security-Policy: frame-ancestors *";
  header("$csp");
  echo "<h3>".$csp."</h3>";
?>

Page 2:
<?php
  $csp= "Content-Security-Policy: frame-ancestors 'self'";
  header("$csp");
  echo "<h3>".$csp."</h3>";
?>
```

```
Page 3:
<?php
  $csp= "Content-Security-Policy: frame-ancestors
  www.example99.com";
  header("$csp");
  echo "<h3>".$csp."</h3>";
?>
```

W5.5. What is the common idea behind the `X-Frame-Options` and CSP mechanisms? Why is it effective in defeating the Clickjacking attack?

W5.6. When a host page puts a page inside an iframe, can the host page access the content inside the iframed page?

W5.7. The following JavaScript code displays content inside a page. The content comes from an untrusted place. If the content, which is supposed to be data only, contains JavaScript code, can the code be executed? Why?

```
const iframe = document.createElement("iframe");
iframe.srcdoc = content;
iframe.sandbox = "";
document.body.appendChild(iframe);
```