TRƯỜNG ĐẠI HỌC KHOA HỌC HUẾ

KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

"Tìm hiểu các kỹ thuật tấn công Clickjacking trên dịch vụ Web"

Học phần: An Ninh Mạng

Giáo viên hướng dẫn : Thầy Võ Việt Dũng

Nhóm sinh viên : Bùi Quang Quý
Trương Thị Hoài Trang
Huỳnh Thị Thủy
Đặng Thị Tâm Nhi

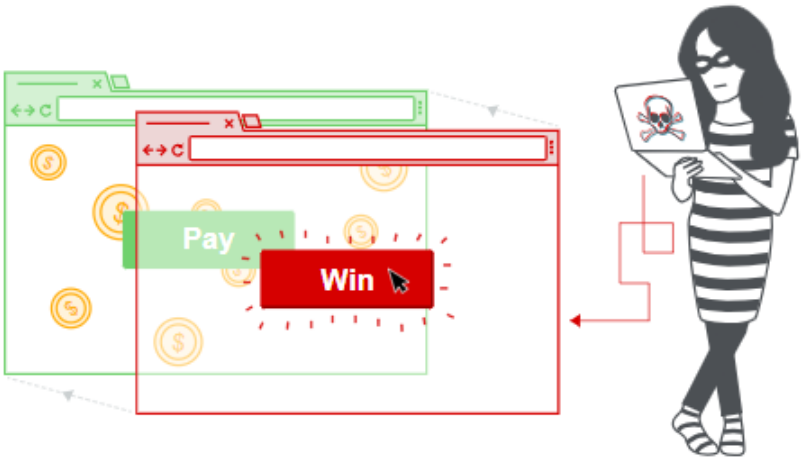
MỤC LỤC

GIỚI THIỆU

An ninh mạng là một lĩnh vực quan trọng trong bối cảnh công nghệ số phát triển mạnh mẽ, với sự gia tăng không ngừng của các mối đe dọa tấn công mạng. Nó bao gồm các biện pháp bảo vệ hệ thống thông tin, dữ liệu và tài nguyên mạng khỏi các cuộc tấn công độc hại nhằm đảm bảo tính toàn vẹn, bảo mật và khả dụng của thông tin. Một trong những phương thức tấn công mạng phổ biến hiện nay là ***Clickjacking*** - một kỹ thuật lừa đảo tinh vi nhằm đánh lừa người dùng thực hiện các hành động không mong muốn trên các trang web. Đề tài này tập trung vào việc nghiên cứu các khái niệm, kỹ thuật tấn công Clickjacking và các phương pháp phòng chống. Thông qua việc tìm hiểu lý thuyết và thực hành thử nghiệm, đề tài không chỉ giúp nâng cao nhận thức về mối đe dọa này mà còn đề xuất các giải pháp hiệu quả để bảo vệ người dùng và hệ thống trước các nguy cơ từ Clickjacking.

I. TỔNG QUAN VỀ CLICKJACKING

1. KHÁI NIỆM

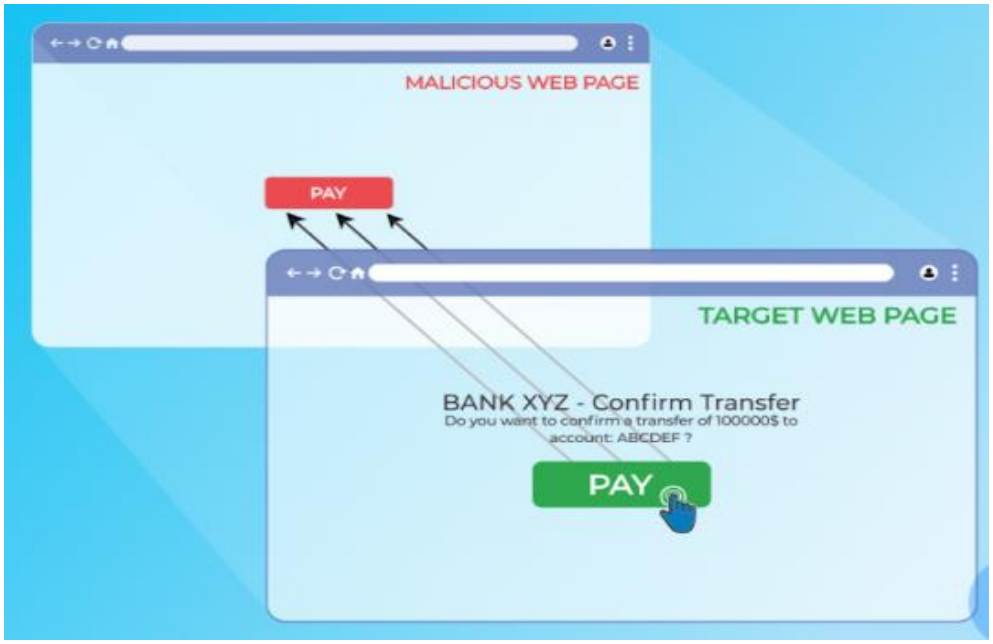


Clickjacking là một hình thức tấn công đánh lừa người dùng nhấp chuột vô ý vào một đối tượng trên website. Khi nhấp chuột vào một đối tượng trên màn hình, người dùng nghĩ là mình đang click vào đối tượng đó nhưng thực chất họ đang bị lừa click vào một đối tượng khác > đã bị làm mờ hay ẩn đi.

2. NGUYÊN LÝ HOẠT ĐỘNG

Clickjacking có thể thực hiện được nhờ các khung HTML hoặc iframe – tức là khả năng hiển thị các trang web trong các trang web khác thông qua các

khung. Về cơ bản, iframe là một khung trong một khung. Iframe cho phép bạn nhúng nội dung từ các nguồn khác vào trang web của mình



- **Tạo lớp phủ (overlay)**: Kẻ tấn công tạo ra một trang web giả mạo hoặc một nội dung hợp pháp, nhưng thực tế họ sẽ đặt một lớp phủ (thường là vô hình) phía trên trang web đó. Lớp phủ này có thể là một iframe chứa nội dung của một trang khác (chẳng hạn như một nút bấm hoặc một form).

- **Lừa người dùng nhấp chuột**: Người dùng nhìn thấy trang web của kẻ tấn công, và họ có thể nghĩ rằng mình đang nhấp vào một phần tử hợp pháp (như một nút hoặc liên kết). Tuy nhiên, thực tế họ đang nhấp vào phần tử của trang web ẩn (như nút thanh toán, like trên mạng xã hội, v.v.).
- **Thực thi hành động không mong muốn**: Do người dùng nhấp vào phần tử ẩn, hành động thực tế sẽ là một hành động không mong muốn mà họ không hề biết. Ví dụ, họ có thể vô tình "like" một trang trên Facebook, thực hiện giao dịch tài chính, hoặc gửi một yêu cầu từ chối mà không biết.

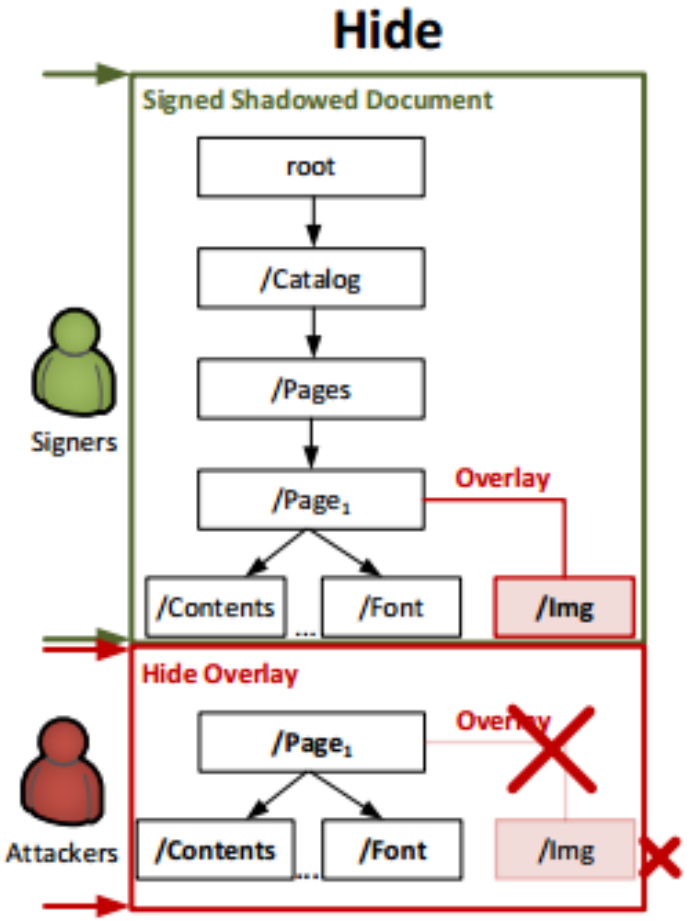
3. CÁC HÌNH THỨC TẤN CÔNG

3.1 Frame Overlay Attack

Frame Overlay Attack là một kiểu tấn công thuộc loại clickjacking, trong đó kẻ tấn công lợi dụng iframe để chồng một trang web độc hại lên một trang web hợp pháp mà người dùng tin cậy.

Trang web độc hại này sẽ có một lớp iframe mỏng, có thể điều chỉnh để che phủ phần trang web mà người dùng muốn tương tác. Kẻ tấn công dùng các kỹ thuật CSS để làm cho iframe không nhìn thấy được hoặc trong suốt.

Khi người dùng cố gắng nhấp vào các nút hoặc liên kết trên trang mà họ tin tưởng, những nhấp chuột này được gửi đến trang web mà kẻ tấn công kiểm



soát, đó là mục đích của kẻ tấn công, nhằm chiếm đoạt thông tin cá nhân, tài khoản của người dùng hoặc thực hiện các giao dịch không mong muốn trên tài khoản của họ mà họ không hề hay biết.

3.2 Likejacking

Likejacking là một hình thức của clickjacking, trong đó kẻ tấn công lợi dụng tính năng "Like" trên mạng xã hội, đặc biệt là Facebook, để lừa người dùng thực hiện hành động không mong muốn-thường là "thích" một trang hoặc bài viết mà họ không thực sự muốn tương tác.

Kẻ tấn công có thể tạo ra một trang web, chứa nội dung giả mạo như video, trò chơi, hoặc khuyến mãi hấp dẫn... nhằm thu hút người dùng. Khi người dùng

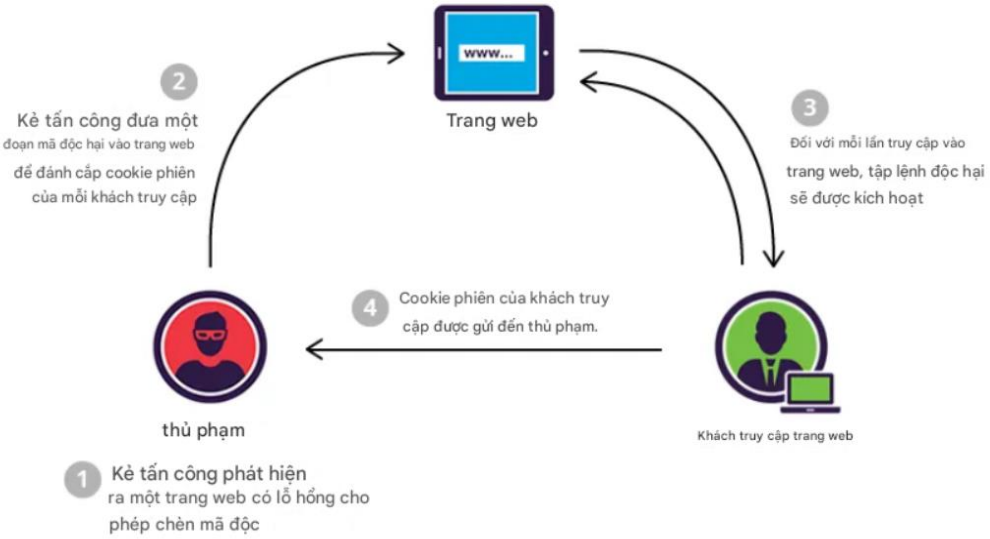
nhấp vào nội dung họ tưởng là hợp lệ, thực chất họ đang nhấp vào nút “Like” ẩn.

Sau hành động đó họ vô tình “Like” một trang hoặc nội dung không mong muốn, thông tin này được hiển thị trên dòng thời gian của họ, và có thể những người bạn của họ thấy hành động này, dẫn đến lây lan thêm nhiều nội dung độc hại, tạo thêm cơ hội cho kẻ tấn công

3.3 UI Redressing

UI Redressing, hay còn gọi là "UI Redress Attack" là một kỹ thuật lừa đảo trong đó kẻ tấn công thay đổi giao diện người dùng của một trang web hoặc ứng dụng để người dùng nhầm lẫn và tương tác với các yếu tố mà họ không dự định. Tấn công này thường được kết hợp với clickjacking và có thể dẫn đến việc tiết lộ thông tin nhạy cảm hoặc thực hiện các hành động không mong muốn.

Mục đích là lừa người dùng thực hiện các hành động không muốn, chẳng hạn như gửi thông tin nhạy cảm, nhấp vào liên kết đến trang web độc hại hoặc chia sẻ nội dung không mong muốn. Tạo ra một trải nghiệm tương tác mà người dùng không nhận ra họ đã bị tấn công. Ví dụ: kẻ tấn công tạo ra một phiên bản giả của một trang web đăng nhập, nhưng thực tế là đang thu thập thông tin đăng nhập của người dùng.



II. PHÂN TÍCH KỸ THUẬT TẤN CÔNG CLICKJACKING

1. CÁC KỸ THUẬT CLICKJACKING

1.1 Kỹ thuật cơ bản- Iframe trong suốt (Transparent Iframe)

Đây là nền tảng của mọi cuộc tấn công Clickjacking. Kẻ tấn công sẽ tạo một trang web "mồi" và nhúng trang web mục tiêu vào một thẻ <iframe> trên trang mồi này. Điểm mấu chốt là kẻ tấn công sẽ sử dụng CSS để làm cho iframe này trở nên trong suốt, khiến người dùng không nhìn thấy nội dung thực sự bên trong.

Cách thức hoạt động:

- Kẻ tấn công tạo một trang web mồi với nội dung hấp dẫn, ví dụ như một trò chơi, một bài kiểm tra, hoặc một thông báo khuyến mãi.
- Họ nhúng trang web mục tiêu vào một thẻ <iframe> trong trang mồi. Ví dụ: <iframe src="https://website-muc-tieu.com"></iframe>
- Bằng cách sử dụng CSS, họ làm cho iframe trở nên trong suốt bằng cách đặt thuộc tính opacity: 0 (hoàn toàn trong suốt) hoặc visibility: hidden (ẩn hoàn toàn). opacity: 0 thường được ưu tiên hơn vì nó vẫn cho phép các sự kiện chuột (như nhấp chuột) được chuyển đến iframe.

Ví dụ

```css
iframe {
  opacity: 0; /* Hoặc visibility: hidden; */
  position: absolute; /* Để định vị iframe chính xác */
  top: 100px;
  left: 50px;
  width: 300px;
  height: 200px;
  z-index: 1; /* Đảm bảo iframe nằm trên các phần tử khác */
}
```

Đoạn mã CSS trên mô tả cách tạo một iframe trong suốt và định vị nó trên trang web, đây chính là cốt lõi của kỹ thuật Clickjacking. Dưới đây là giải thích chi tiết từng thuộc tính:

- **opacity: 0;**: Thuộc tính này làm cho iframe hoàn toàn trong suốt. Người dùng sẽ không thể nhìn thấy nội dung của iframe trên trang web. Giá trị 0 nghĩa là hoàn toàn trong suốt
- **position: absolute;**: Thuộc tính này thiết lập kiểu định vị của iframe là tuyệt đối. cho phép bạn đặt iframe ở bất kỳ vị trí nào trên trang web một cách chính xác.

1.2 Chỉ số Z (Z-index)

Z-index là một thuộc tính CSS trong thiết kế web dùng để xác định thứ tự chồng lấp (stacking order) của các phần tử trên trang web. Giá trị của Z-index càng lớn, phần tử càng nằm ở phía trên cùng trong thứ tự chồng lấp, và ngược lại. Phần tử không có Z-index hoặc có giá trị Z-index thấp hơn sẽ bị che bởi các phần tử có giá trị cao hơn.Kẻ tấn công lợi dụng **Z-index** để đặt một khung iframe chứa nội dung hợp pháp (ví dụ: biểu mẫu đăng nhập) ở phía dưới. Đặt một lớp che (thường là nút hoặc giao diện độc hại) ở phía trên, với Z-index cao hơn.. Làm lớp giao diện độc hại này trong suốt hoặc được thiết kế sao cho người dùng không nhận biết được sự tồn tại của nó.

Cách thức hoạt động

Trong Clickjacking**, z-index** đóng vai trò *quyết định* trong việc che giấu iframe chứa nội dung độc hại. Kẻ tấn công cần đảm bảo iframe

(đã được làm trong suốt bằng opacity: 0) nằm *ngay trên* phần tử mà người dùng nhìn thấy và tương tác (phần tử "mồi"). Đây là cách **z-index** được sử dụng:

- **Phần tử mồi :** Đây là phần tử mà người dùng *nhìn thấy* và được thiết kế để dụ dỗ họ nhấp chuột. Ví dụ: một nút "Nhận quà", một liên kết "Xem video", v.v. Phần tử này thường có position: relative (định vị tương đối-tạo ngữ cảnh xếp chồng); hoặc position: absolute;(định vị tuyệt đối- định vị cho iframe).
- **Iframe độc hại (hidden iframe):** Đây là iframe chứa trang web mục tiêu mà kẻ tấn công muốn người dùng tương tác một cách vô thức. Iframe này *bắt buộc* phải có position: absolute; để có thể được định vị chính xác và được xếp chồng lên phần tử mồi.
- **z-index quyết định thứ tự:**
  - Phần tử mồi được gán một giá trị z-index thấp hơn (ví dụ: z-index: 1; hoặc thậm chí không cần gán z-index nếu các phần tử khác cũng không có).
  - Iframe độc hại được gán một giá trị z-index *cao hơn* (ví dụ: z-index: 2;).

1.3 Các kỹ thuật nâng cao

- **Cursorjacking:** Kỹ thuật này thay đổi hình dạng hoặc vị trí của con trỏ chuột để đánh lừa người dùng. Ví dụ: con trỏ chuột có thể bị ẩn đi và một con trỏ giả được hiển thị ở một vị trí khác, khiến người dùng nhấp vào một vị trí khác với những gì họ nghĩ. Điều này thường được thực hiện bằng JavaScript.
- **Quick jacking (hoặc Rapid Clickjacking):** Sử dụng các hiệu ứng động hoặc chuyển động nhanh để làm người dùng khó nhận biết được sự hiện diện của iframe ẩn. Ví dụ: iframe có thể chỉ hiển thị trong một khoảng thời gian rất ngắn (vài mili giây).
- **Clickjacking kết hợp với XSS (Cross-Site Scripting):** Mặc dù ít phổ biến hơn do các biện pháp bảo mật XSS ngày càng được cải thiện, kẻ tấn công có thể lợi dụng lỗ hổng XSS trên trang web mục tiêu để chèn iframe độc hại một cách trực tiếp. Điều này bỏ qua nhu cầu tạo trang web mồi riêng, vì iframe được chèn trực tiếp vào trang web bị tấn công.

2. MÔ PHỎNG KỸ THUẬT TẤN CÔNG CƠ BẢN

Để demo, chúng ta cần hai trang web:

- **Trang web mục tiêu (victim.html):** Đây là trang web mà chúng ta muốn người dùng tương tác "một cách vô ý". Ví dụ, một nút "Xác nhận chuyển tiền" hoặc "Like fanpage".
- **Trang web tấn công (attacker.html):** Đây là trang web mà người dùng nhìn thấy và tương tác. Nó chứa iframe trong suốt che phủ lên một phần của trang web

❖ **Mã nguồn**

victim.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Trang web mục tiêu</title>
</head>
<body>
   <h1>Đây là trang web mục tiêu</h1>
   <button id="transferButton">Xác nhận chuyển tiền</button>
   <script>
      document.getElementById("transferButton").addEventListener("click",
function() {
         alert("Đã chuyển tiền!");
      });
   </script>
</body>
</html>
```

attacker.html:

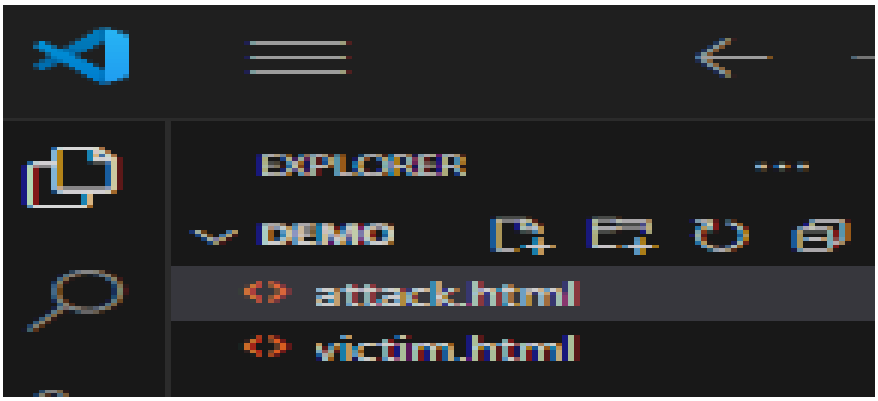
```
<!DOCTYPE html>
<html>
<head>
   <title>Trang web tấn công</title>
   <style>
      #overlay {
         position: absolute;
         top: 50px; /* Điều chỉnh vị trí */
         left: 100px; * Điều chỉnh vị trí */
         opacity: 0.5; /* Độ trong suốt (để dễ nhìn trong demo, thường là 0) */
         z-index: 1; /* Đảm bảo iframe nằm trên cùng */
      }
   </style>
</head>
<body>
   <h1>Đây là trang web mà bạn nhìn thấy</h1>
   <p>Nhấp vào đây để nhận quà!</p>
   <iframe id="overlay" src="victim.html" width="200" height="50"></iframe>
</body>
</html>
```

Giải thích:

- Trong attacker.html, chúng ta tạo một iframe với id="overlay" và đặt thuộc tính src là victim.html.
- CSS được sử dụng để định vị iframe bằng position: absolute, top và left.
- Quan trọng nhất là opacity: 0.5. Trong tấn công thực tế, giá trị này sẽ là 0 để iframe hoàn toàn trong suốt. trong demo đặt giá trị này là 0.5 để có thể thấy iframe trong quá trình demo.
- z-index: 1 đảm bảo iframe nằm trên cùng của trang web.

❖ **Quá trình demo**

Bước 1: tạo và lưu hai file `victim.html` và `attacker.html` vào cùng một thư mục



Mở victim.html bằng trình duyệt ta sẽ thấy nội dung hiển thị

Đây là trang web mục tiêu

Xác nhận chuyển tiền

Bước 2 Mở attacker.html bằng trình duyệt

Đây là trang web mà bạn nhìn thấy

Nhấp vào đây để nhận quà!

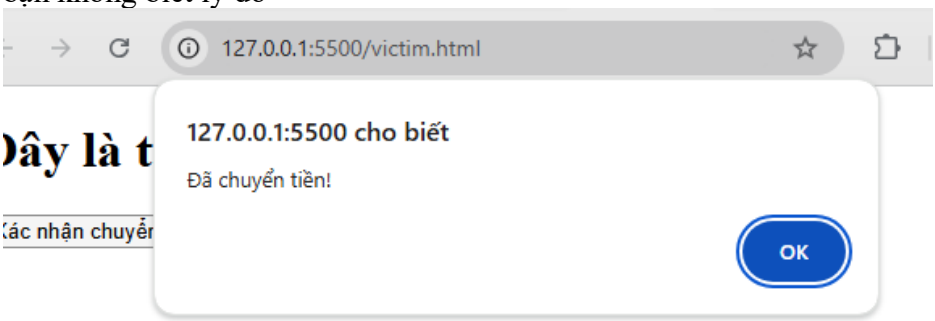
attacker.html

Trình duyệt sẽ hiện thị như trên . Bạn sẽ thấy trang "Đây là trang web mà bạn nhìn thấy" và một phần của trang "Đây là trang web mục tiêu" (nút "**Xác nhận chuyển tiền")** bị che phủ bởi iframe trong suốt. thay vào đó chúng ta chỉ thấy

nút "**nhấp vào đây để nhận quà !**" nhưng thực chất nó là nút xác nhận chuyển tiền

\

Khi bạn nhấp vào bất kỳ đâu trong khu vực iframe, bạn thực sự đang nhấp vào nút "**Xác nhận chuyển tiền**" trên `victim.html`. Trong demo này, một hộp thoại alert "Đã chuyển tiền!" sẽ hiện lên và tiền của bạn đã bị chuyển đi mà bạn không biết lý do



III. CÁCH PHÒNG CHỐNG

1. CÁC BIỆN PHÁP BẢO MẬT

1.1 Từ phía người dùng(Client)

Mặc dù người dùng không thể ngăn chặn hoàn toàn các cuộc tấn công nhưng họ có thể sử dụng các biện pháp phòng ngừa để giảm thiểu rủi ro.

- *Cẩn trọng các liên kết(be cautious of links)*

  Kẻ tấn công thường sử dụng các kỹ thuật lừa đảo để dụ người dùng nhấp vào các liên kết độc hại. Bản phải kiểm tra kỹ URL trước khi nhấp vào. Không nhấp vào các liên kết không tin cậy. Và hãy cẩn thận với các pop-up hoặc các quảng cáo đột ngột

- *Cập nhật trình duyệt và phần mềm (update browsers and software).*
  Các phiên bản cũ của trình duyệt và phầm mềm có thể chứa các lỗ hổng bảo mật mà kẻ tấn công có thể khai thác. Hãy

bật tính năng tự động cập nhật cho trình duyệt cũng như hệ điều hành.

- *Sử dụng tiện ích mở rộng(use security extensions)*

  Một số tiện ích mở rộng của trình duyệt có thể giúp phát hiện và ngăn chặn các cuộc tấn công clickjacking bằng cách cảnh báo người dùng về các iframe ẩn hoặc các hành vi đáng ngờ (Ví dụ: NoScript cho FireFox là một tiện ích mở rộng cho phép người dùng kiểm tra việc thực thi javaScript trên các trang web giúp ngăn chặn một số kỹ thuật clickjacking)

1.2 Từ phía máy chủ (Server)

1.2.1 Sử dụng X-Frame-Options

- **Khái niệm:** *x-Frame-Options* là một tiêu đề bảo mật HTTP thiết yếu được thiết kế để bảo vệ các ứng dụng web khỏi bị nhúng trong iframe trên các trang web trái phép. Bằng cách thiết lập tiêu đề này, nhằm ngăn các trang web khác đóng khung nội dung của mình, giúp giảm thiểu các rủi ro
- **Nguyên lý hoạt động** X-Frame-Options hỗ trợ ba chỉ thị, mỗi chỉ thị cung cấp một mức độ bảo vệ khác nhau:
  - **DENY** : Lệnh deny của X-Frame-Options ngăn chặn hoàn toàn nội dung của bạn được hiển thị trong bất kỳ khung hoặc iframe nào, bất kể nguồn gốc (*ngăn chặn hoàn toàn việc nhúng trang web vào bất kì iframe*)
  - **SAMEORIGIN** :chỉ cho phép nội dung của bạn được hiển thị trong một khung hoặc iframe chỉ khi yêu cầu đến từ cùng một nguồn gốc với trang web của bạn. Điều này có nghĩa là trang web của bạn có thể được đóng khung bởi các trang khác trên cùng một tên miền nhưng không phải bởi các trang từ các tên miền khác. *(chỉ cho nhúng trang web vào iframe chỉ khi cùng nguồn gốc như domain, protocol, port)*
  - **ALLOW-FROM uri** : Chỉ thị này cho phép bạn chỉ định một nguồn gốc duy nhất được phép đóng khung nội dung của bạn. Tuy nhiên, chỉ thị này đã lỗi thời và không được hỗ trợ rộng rãi, vì vậy, nói chung, nên sử dụng chỉ thị `Content-Security-Policy` frame-ancestors để linh hoạt hơn.*(chỉ cho phép nhúng trang web vào iframe từ một URL cụ thể*)

1.2.2 Content Security Policy (CSP)

- **Khái niệm :** CSP giống như một danh sách kiểm soát nghiêm ngặt cho trình duyệt, quy định rõ ràng những gì được phép và không được phép tải trên trang web. Nếu một tài nguyên (ví dụ: script, style, image, iframe) không nằm trong danh sách được phê duyệt, trình duyệt sẽ chặn nó.

- **Nguyên lý hoạt động: k**hi trình duyệt nhận được một trang web, nó sẽ phân tích header Content-Security-Policy. Header này chứa các "chỉ thị" (directives) quy định các nguồn hợp lệ cho từng loại tài nguyên. Ví dụ:

  - script-src: Quy định các nguồn hợp lệ cho JavaScript.

  - style-src: Quy định các nguồn hợp lệ cho CSS.

  - img-src: Quy định các nguồn hợp lệ cho hình ảnh.

  - frame-ancestors: Quy định các nguồn hợp lệ cho việc nhúng trang web vào iframe.

Nếu trình duyệt gặp một tài nguyên không tuân theo các chỉ thị này, nó sẽ chặn tài nguyên đó và có thể báo cáo vi phạm (nếu được cấu hình).

1.2.3 Frame Busting Scripts

- **Khái niệm:** Frame Busting Scripts (hay còn gọi là Framebusting) là các đoạn mã JavaScript được sử dụng trên một trang web để ngăn chặn trang web đó bị nhúng vào bên trong một <iframe> trên một trang web khác
- **Nguyên lý hoạt động:** Nguyên lý cơ bản của Frame Busting Scripts là kiểm tra xem trang web hiện tại có đang nằm trong một iframe hay không. Nếu có, script sẽ thực hiện một hành động để "thoát ra" khỏi iframe và hiển thị trang web ở cửa sổ trình duyệt đầy đủ.

1.2.4 Double Confirm

- **Khái niệm :**Double Confirm" hay "Double Confirmation" (xác nhận kép) là một kỹ thuật bảo mật, trong đó người dùng được yêu cầu thực hiện một hành động *hai lần* để xác nhận ý định của họ. Mục đích là để giảm thiểu rủi ro từ các cuộc tấn công lừa đảo như Clickjacking, trong đó người dùng có thể vô tình thực hiện một hành động mà họ không hề hay biết.

- **Nguyên lý hoạt động:** Nguyên lý rất đơn giản thay vì chỉ có một nút "Xác nhận" duy nhất, người dùng sẽ phải thực hiện hai bước riêng biệt để hoàn thành một hành động quan trọng.

  Ví dụ:

  Bước 1: Hành động ban đầu: Người dùng nhấp vào một nút hoặc liên kết để bắt đầu một hành động (ví dụ: "Xóa tài khoản", "Chuyển tiền", "Mua hàng").

  Bước 2: Xác nhận thứ hai: Một hộp thoại hoặc một trang mới xuất hiện, yêu cầu người dùng xác nhận lại hành động

của họ. Hộp thoại này thường chứa thông tin chi tiết về hành động sẽ được thực hiện và một nút "Xác nhận" thứ hai.

Double Confirm giúp chống Clickjacking bằng cách làm cho việc tấn công trở nên khó khăn hơn đáng kể. Kẻ tấn công cần phải tạo ra một tình huống phức tạp hơn để lừa người dùng nhấp chuột hai lần vào đúng vị trí trên iframe ẩn.

2. ĐÁNH GIÁ

Phương pháp	Ưu điểm	Nhược điểm	Khuyến nghị
Frame Busting	Dễ triển khai, có thể hoạt động trên trình duyệt cũ.	Kém hiệu quả, dễ bị vượt qua, bị vô hiệu hóa bởi XFO/CSP.	KHÔNG NÊN SỬ DỤNG.
X-Frame-Options	Đơn giản, dễ cấu hình, hiệu quả, được hỗ trợ rộng rãi.	Hạn chế về khả năng kiểm soát, không linh hoạt bằng CSP.	TỐT nếu chỉ cần giải pháp đơn giản.
CSP với `frame-ancestors`	Kiểm soát chi tiết, linh hoạt, mạnh mẽ, được khuyến nghị nhất hiện nay.	Cấu hình phức tạp hơn XFO, yêu cầu trình duyệt hỗ trợ CSP.	TỐT NHẤT.
Double Confirm	Tăng cường bảo mật, tránh hành động vô ý.	Gây bất tiện cho người dùng, không hoàn hảo, chỉ là biện pháp bổ sung.	Nên được sử dụng KẾT HỢP với XFO hoặc CSP.

IV. KẾT LUẬN VÀ TÀI LIỆU THAM KHẢO

Clickjacking là một kiểu tấn công nguy hiểm lợi dụng iframe để đánh lừa người dùng nhấp vào những nội dung ẩn. Để phòng chống tấn công này, có bốn phương pháp chính với mức độ hiệu quả khác nhau. Frame Busting Scripts, mặc dù dễ triển khai, đã lỗi thời và dễ bị qua mặt. X-Frame-Options là một giải pháp đơn giản và hiệu quả, đặc biệt với tùy chọn DENY hoặc SAMEORIGIN, nhưng lại kém linh hoạt hơn so với CSP. Content Security Policy (CSP) với chỉ thị **frame-ancestors***(là một phần của CSP sử dụng để kiểm soát việc trang web có thể được nhúng vào iframe trên trang web nào)* là phương pháp mạnh mẽ và linh hoạt nhất, cho phép kiểm soát chi tiết các nguồn được phép nhúng trang web, và được khuyến nghị sử dụng nhất hiện nay. Cuối cùng, Double Confirm (Xác nhận kép) là một lớp bảo vệ bổ sung, giúp người dùng tránh các hành động vô ý bằng cách yêu cầu xác nhận lại, nhưng không hoàn toàn ngăn chặn được tấn công. Vì vậy, để bảo vệ tối ưu, nên ưu tiên sử dụng CSP với **frame-ancestors**, nếu không khả thi thì dùng X-Frame-Options, kết hợp với Double Confirm cho các hành động quan trọng. Việc cập nhật kiến thức và áp dụng các biện pháp bảo mật mới là rất quan trọng để đảm bảo an toàn

cho trang web và người dùng. Và trên hết là người dùng phải nâng cao cách giác khi tham gia vào trình duyệt.

TÀI LIỆU THAM KHẢO

1. **https://www.indusface.com/learning/x-frame-options/?amp**
2. **https://auth0.com/blog/preventing-clickjacking-attacks/**
3. **https://portswigger.net/web-security/clickjacking**
4. **https://www.indusface.com/blog/what-are-clickjacking-attacks-tips-to-prevent-them/**
5. **https://www.getastra.com/blog/cms/php-security/prevent-clickjacking-in-php/**