Overview:

Giả mạo yêu cầu chéo trang web (còn được gọi là CSRF) là một lỗ hổng bảo mật web cho phép kẻ tấn công gây ra người dùng thực hiện các hành động mà họ không có ý định thực hiện. Nó cho phép kẻ tấn công phá vỡ một phần chính sách nguồn gốc tương tự, được thiết kế để ngăn chặn các trang web khác nhau can thiệp lẫn nhau.

Impact:

Trong một cuộc tấn công CSRF thành công, kẻ tấn công khiến người dùng nạn nhân thực hiện một hành động vô tình. Ví dụ: điều này có thể là thay đổi địa chỉ email trên tài khoản của họ, để thay đổi mật khẩu của họ hoặc thực hiện chuyển tiền. Tùy thuộc vào bản chất của hành động, kẻ tấn công có thể giành toàn quyền kiểm soát tài khoản của người dùng. Nếu người dùng bị xâm phạm có vai trò đặc quyền trong ứng dụng, thì kẻ tấn công có thể kiểm soát hoàn toàn tất cả dữ liệu và chức năng của ứng dụng.

Activate:

Để CSRF hoạt động được cần phải có 3 điều kiện bắt buộc:

- **A relevant action:** Có một hành động trong ứng dụng mà kẻ tấn công có lý do để gây ra. Đây có thể là một hành động đặc quyền (chẳng hạn như sửa đổi quyền cho người dùng khác) hoặc bất kỳ hành động nào đối với dữ liệu dành riêng cho người dùng (chẳng hạn như thay đổi mật khẩu của chính người dùng).
- **Cookie-based session handling:** Thực hiện hành động liên quan đến việc phát hành một hoặc nhiều yêu cầu HTTP và ứng dụng chỉ dựa vào cookie phiên để xác định người dùng đã thực hiện yêu cầu. Không có cơ chế nào khác để theo dõi phiên hoặc xác thực yêu cầu của người dùng.
- **No unpredictable request parameters:** Các yêu cầu thực hiện hành động không chứa bất kỳ tham số nào có giá trị kẻ không thể xác định hoặc đoán. Ví dụ: khi khiến người dùng thay đổi mật khẩu của họ,

hàm không dễ bị tổn thương nếu kẻ tấn công cần biết giá trị của mật khẩu hiện có.

Types:

- **Validation of CSRF token depends on request method:** Một số ứng dụng xác thực chính xác mã thông báo khi yêu cầu sử dụng phương pháp POST nhưng bỏ qua xác thực khi phương pháp GET được sử dụng.Trong trường hợp này, kẻ tấn công có thể chuyển sang phương pháp GET để bỏ qua xác nhận và cung cấp một cuộc tấn công CSRF:
- **Validation of CSRF token depends on token being present:** Một số ứng dụng xác thực chính xác mã thông báo khi có mặt nhưng bỏ qua xác thực nếu mã thông báo bị bỏ qua.Trong trường hợp này, kẻ tấn công có thể loại bỏ toàn bộ tham số chứa mã thông báo (không chỉ giá trị của nó) để bỏ qua xác thực và cung cấp một cuộc tấn công CSRF
- **CSRF token is not tied to the user session:** Một số ứng dụng không xác thực rằng mã thông báo thuộc cùng một phiên với người dùng đang thực hiện yêu cầu. Thay vào đó, ứng dụng duy trì một nhóm mã thông báo toàn cầu mà nó đã phát hành và chấp nhận bất kỳ mã thông báo nào xuất hiện trong nhóm này.Trong trường hợp này, kẻ tấn công có thể đăng nhập vào ứng dụng bằng tài khoản của riêng họ, có được một mã thông báo hợp lệ, và sau đó cung cấp mã thông báo đó cho người dùng nạn nhân trong cuộc tấn công CSRF của họ.
- **CSRF token is tied to a non-session cookie**: Trong một biến thể trên lỗ hổng trước đó, một số ứng dụng gắn mã thông báo CSRF với cookie, nhưng không phải với cùng một cookie được sử dụng để theo dõi các phiên. Điều này có thể dễ dàng xảy ra khi một ứng dụng sử dụng hai khung khác nhau, một để xử lý phiên và một để bảo vệ CSRF, không được tích hợp với nhau
- **CSRF token is simply duplicated in a cookie**: Trong một biến thể khác trên lỗ hổng trước đó, một số ứng dụng không duy trì bất kỳ bản ghi phía máy chủ nào của mã thông báo đã được phát hành, mà thay vào đó sao chép từng mã thông báo trong cookie và tham số yêu cầu. Khi yêu cầu tiếp theo được xác thực, ứng dụng chỉ cần xác minh rằng mã thông báo được gửi trong tham số yêu cầu khớp với giá trị được gửi trong cookie. Điều này đôi khi được gọi là phòng thủ "đệ trình kép" chống lại CSRF và được ủng hộ vì nó rất đơn giản để thực hiện và tránh sự cần thiết của bất kỳ trạng thái phía máy chủ nào.

- **Referer-based defenses against CSRF:** Ngoài các biện pháp phòng thủ sử dụng mã thông báo CSRF, một số ứng dụng sử dụng tiêu đề HTTP để cố gắng bảo vệ chống lại các cuộc tấn công CSRF, thông thường bằng cách xác minh rằng yêu cầu có nguồn gốc từ tên miền riêng của ứng dụng. Cách tiếp cận này thường kém hiệu quả hơn và thường bị bỏ qua. Referer
- **Validation of Referer depends on header being present:** Một số ứng dụng xác thực tiêu đề khi nó có trong các yêu cầu nhưng bỏ qua xác thực nếu tiêu đề bị bỏ qua. RefererTrong trường hợp này, kẻ tấn công có thể tạo csrf khai thác của họ theo cách khiến trình duyệt của người dùng nạn nhân thả tiêu đề trong yêu cầu kết quả. Có nhiều cách khác nhau để đạt được điều này, nhưng cách dễ nhất là sử dụng thẻ META trong trang HTML lưu trữ cuộc tấn công CSRF
- **Validation of Referer can be circumvented**: Một số ứng dụng xác thực tiêu đề theo cách ngây thơ có thể được bỏ qua. Ví dụ: nếu ứng dụng xác nhận rằng tên miền trong bắt đầu với giá trị mong đợi, thì kẻ tấn công có thể đặt tên miền phụ này làm tên miền phụ của tên miền riêng của họ: RefererReferer.Tương tự như vậy, nếu ứng dụng chỉ đơn giản là xác thực rằng chứa tên miền của riêng mình, thì kẻ tấn công có thể đặt giá trị cần thiết ở nơi khác trong URL: Referer

Protect:

Để bảo vệ khỏi các cuộc tấn công CSRF thì ta cần phải đòi hỏi 2 điều:

- Đảm bảo rằng các yêu cầu GET không có tác dụng phụ
- Đảm bảo rằng các yêu cầu không phải GET chỉ có thể được bắt nguồn từ mã phía máy khách

*** REST (**Representation State Transfer**):** là một loạt các nguyên tắc thiết kế gán một số loại hành động nhất định (xem, tạo, xóa, cập nhật) cho các động từ HTTP khác nhau. Theo các thiết kế REST-ful sẽ giữ cho mã của bạn sạch sẽ và giúp quy mô trang web của bạn. Hơn nữa, REST khẳng định rằng các yêu cầu GET chỉ được sử dụng để xem tài nguyên. Giữ cho yêu cầu GET của bạn không có tác dụng phụ sẽ hạn chế tác hại có thể được thực hiện bởi các URL được tạo ra độc hại - kẻ tấn công sẽ phải làm việc chăm chỉ hơn nhiều để tạo ra các yêu cầu POST có hại.

*__Anti-Forgery Tokens__: Mỗi khi máy chủ của bạn hiển thị một trang thực hiện các hành động nhạy cảm, nó sẽ viết ra một mã thông báo chống giả mạo trong trường biểu mẫu HTML ẩn. Máy chủ sẽ xác thực mã thông báo khi nó được trả về trong các yêu cầu tiếp theo và từ chối bất kỳ cuộc gọi nào với mã thông báo bị thiếu hoặc không hợp lệ. Mã thông báo chống giả mạo thường là (mạnh) các số ngẫu nhiên được lưu trữ trong cookie hoặc trên máy chủ khi chúng được ghi vào trường ẩn. Máy chủ sẽ so sánh mã thông báo được đính kèm với yêu cầu đến với giá trị được lưu trữ trong cookie. Nếu các giá trị giống hệt nhau, máy chủ sẽ chấp nhận yêu cầu HTTP hợp lệ.

***Ensure Cookies are sent with the SameSite Cookie Attribute:**

***Include Addition Authentication for Sensitive Actions:** Nhiều trang web yêu cầu bước xác thực phụ hoặc yêu cầu xác nhận lại chi tiết đăng nhập khi người dùng thực hiện hành động nhạy cảm. (Hãy nghĩ đến trang đặt lại mật khẩu điển hình - thông thường người dùng sẽ phải chỉ định mật khẩu cũ của họ trước khi đặt mật khẩu mới.) Điều này không chỉ bảo vệ người dùng có thể vô tình đăng nhập vào các máy tính có thể truy cập công khai mà còn làm giảm đáng kể khả năng tấn công CSRF.

Reference

- https://www.hacksplaining.com/prevention/csrf
- https://portswigger.net/web-security/csrf
- https://owasp.org/www-community/attacks/csrf