**Tìm hiểu CSRF là gì? Cách phòng chống tấn công giả mạo**

[Security](https://viblo.asia/tags/security)[CSRF](https://viblo.asia/tags/csrf)

Hôm nay mình mời các bạn đi tìm hiểu tìm hiểu CSRF (Cross-site Request Forgery) là gì, qua cách phòng chống tấn công giả mạo qua bài viết dưới đây.

**Vậy CSRF (Cross-site Request Forgery) là gì?**

CSRF (Cross-site Request Forgery) nói đến việc tấn công vào chứng thực request trên web thông qua việc sử dụng Cookies, nơi mà các hacker có khả năng sử dụng thủ thuật để tạo request mà bạn không hề biết. Vì vậy, một CSRF là hacker lạm dụng sự tin tưởng của một ứng dụng web trên trình duyệt của nạn nhân.

CSRF là một kiểu tấn công gây sự nhầm lẫn tăng tính xác thực và cấp quyền của nạn nhân khi gửi một request giả mạo đến máy chủ. Vì thế một lỗ hổng CSRF ảnh hưởng đến các quyền của người dùng ví dụ như quản trị viên, kết quả là chúng truy cập được đầy đủ quyền.

Trong khi một tấn công CSRF trình duyệt của nạn nhân bị lừa để gửi đi các request đến ứng dụng web theo mong muốn của kẻ tấn công, thông thường, ví dụ một yêu cầu sẽ được gửi lên để thay đổi một vài dữ liệu phía server.



Khi gửi một request HTTP (hợp pháp hoặc cách khác), trình duyệt của nận nhân sẽ bao gồm các tiêu đề cookie. Các cookie thường được dùng để lưu trữ một session để định danh người dùng không phải xác thực lại cho mỗi yêu cầu gửi lên, điều này hiển nhiên là không thực tế.

Nếu phiên làm việc đã xác thực của nạn nhân được lưu trữ trong một Cookie vẫn còn hiệu lực (một cửa sổ trình duyệt không nhất thiết phải mở), và nếu ứng dụng dễ bị tấn công CSRF, kẻ tấn công có thể thử dụng CSRF để chay bất cứ yêu cầu nào với trang web mà trang web không thể phân biệt được request nào là hợp pháp hay không.

**Lịch sử về tấn công CSRF**

Các kiểu tấn công CSRF xuất hiện từ những năm 1990, tuy nhiên các cuộc tấn công này xuất phát từ chính IP của người sử dụng nên log file của các website k cho thấy các dấu hiệu của CSRF. Các cuộc tấn công theo kĩ thuật CSRF k dc báo cáo đầy đủ, đến năm 2007 mới có một vài tài liệu miêu tả chi tiết về các trường hợp tấn công CSRF.

Năm 2008 người ta phát hiện ra có khoảng 18 triệu người sử dụng eBay ở Hàn Quốc mất các thông tin cá nhân của mình. Cũng trong năm 2008, một số khách hàng tại ngân hàng Mexico bị mất tài khoản cá nhân của mình.Trong 2 trường hợp kể trên hacker đều sử dụng kĩ thuật tấn công CSRF.

**Kịch bản tấn công CSRF**

Bạn có thể xem xét ví dụ sau để hiểu hơn về tấn công CSRF:

Kẻ tấn công là Alice chọn mục tiêu là chiếc ví của Todd bằng cách chuyển một phần tiền của Todd cho cô ta. Ngân hàng của Todd đã gặp phải lỗ hổng CSRF. Để gửi tiền, Todd phải truy cập vào URL sau:

<http://example.com/app/transferFunds?amount=1500&destinationAccount=4673243243>

Sau khi URL này được mở ra, một trang thành công được trình bày cho Todd và việc chuyển đổi đã hoàn tất. Alice cũng biết rằng Todd thường ghé thăm một trang web dưới quyền kiểm soát của cô tại [blog.aliceisawesome.com](http://blog.aliceisawesome.com/), nơi cô đặt đoạn mã sau đây:

<img src = "http://example.com/app/transferFunds?amount=1500&destinationAccount=4673243243" width = "0" height = "0" />

Khi truy cập trang web của Alice, trình duyệt của Todd nghĩ rằng Alice liên kết đến một hình ảnh và tự động đưa ra yêu cầu HTTP GET để lấy “hình ảnh”, nhưng điều này thực sự hướng dẫn ngân hàng của Todd chuyển $1500 đến Alice.

Cách phòng chống tấn công giả mạo

Dựa trên nguyên tắc của CSRF “lừa trình duyệt của người dùng (hoặc người dùng) gửi các câu lệnh HTTP”, các kĩ thuật phòng tránh sẽ tập trung vào việc tìm cách phân biệt và hạn chế các câu lệnh giả mạo.



**Phía User**

Để phòng tránh trở thành nạn nhân của các cuộc tấn công CSRF, người dùng internet nên thực hiện một số lưu ý sau:

* Nên thoát khỏi các website quan trọng: Tài khoản ngân hàng, thanh toán trực tuyến, các mạng xã hội, gmail, yahoo… khi đã thực hiện xong giao dịch hay các công việc cần làm. (Check – email, checkin…)
* Không nên click vào các đường dẫn mà bạn nhận được qua email, qua facebook … Khi bạn đưa chuột qua 1 đường dẫn, phía dưới bên trái của trình duyệt thường có địa chỉ website đích, bạn nên lưu ý để đến đúng trang mình muốn.
* Không lưu các thông tin về mật khẩu tại trình duyệt của mình (không nên chọn các phương thức “đăng nhập lần sau”, “lưu mật khẩu” …
* Trong quá trình thực hiện giao dịch hay vào các website quan trọng không nên vào các website khác, có thể chứa các mã khai thác của kẻ tấn công.

**Phía Server**

Có nhiều lời khuyến cáo được đưa ra, tuy nhiên cho đến nay vẫn chưa có biện pháp nào có thể phòng chống triệt để CSRF. Sau đây là một vài kĩ thuật sử dụng.

* Sử dụng GET và POST đúng cách: Dùng GET nếu thao tác là truy vấn dữ liệu. Dùng POST nếu các thao tác tạo ra sự thay đổi hệ thống. Nếu ứng dụng của bạn theo chuẩn RESTful, bạn có thể dùng thêm các HTTP verbs, như PATCH, PUThay DELETE
* Sử dụng captcha, các thông báo xác nhận: Captcha được sử dụng để nhận biết đối tượng đang thao tác với hệ thống là con người hay không? Các thao tác quan trọng như “đăng nhập” hay là “chuyển khoản” ,”thanh toán” thường là hay sử dụng captcha. Tuy nhiên, việc sử dụng captcha có thể gây khó khăn cho một vài đối tượng người dùng và làm họ khó chịu. Các thông báo xác nhận cũng thường được sử dụng, ví dụ như việc hiển thị một thông báo xác nhận “bạn có muốn xóa hay không” cũng làm hạn chế các kĩ thuật Cả hai cách trên vẫn có thể bị vượt qua nếu kẻ tấn công có một kịch bản hoàn hảo và kết hợp với lỗi XSS.
* Sử dụng token: Tạo ra một token tương ứng với mỗi form, token này sẽ là duy nhất đối với mỗi form và thường thì hàm tạo ra token này sẽ nhận đối số là “SESSION” hoặc được lưu thông tin trong SESSION. Khi nhận lệnh HTTP POST về, hệ thống sẽ thực hiên so khớp giá trị token này để quyết định có thực hiện hay không.
* Sử dụng cookie riêng biệt cho trang quản trị: Một cookie không thể dùng chung cho các domain khác nhau,chính vì vậy việc sử dụng “[admin.site.com](http://admin.site.com/)” thay vì sử dụng “[site.com/admin”](http://site.com/admin%E2%80%9D) là an toàn hơn.
* Kiểm tra REFERRER: Kiểm tra xem các câu lệnh http gửi đến hệ thống xuất phát từ đâu. Một ứng dụng web có thể hạn chế chỉ thực hiện các lệnh http gửi đến từ các trang đã được chứng thực. Tuy nhiên cách làm này có nhiều hạn chế và không thật sự hiệu quả.
* Kiểm tra IP: Một số hệ thống quan trọng chỉ cho truy cập từ những IP được thiết lập sẵn.