Nền tảng phát triển Web

Các vấn đề về bảo mật trong web

thongph88@gmail.com

Nội dung

1) Khái niệm
2) Tại sao lại phải bảo mật trong web
3) HTTPS
4) Same-Origin Policy – SOP
5) CORS
6) Web tracking
7) Cross-Site Request Forgery (CSRF)
8) Tấn công Cross-Site Scripting (XSS)
9) SQL injection

Khái niệm

- Bảo mật website là một chức năng, nhiệm vụ vô cùng thiết yếu đảm bảo tính an toàn cho website trong quá trình vận hành và sử dụng.
- Các nhà quản trị nên thường xuyên kiểm tra và xây dựng hệ thống bảo mật cấp cao để tránh khỏi bất cứ điều gì có thể xảy ra khi hacker tấn công.
- Để sở hữu một website vận hành tốt, trơn tru, hãy chắc rằng bạn đã và đang bảo mật website của mình theo một cách tốt nhất.

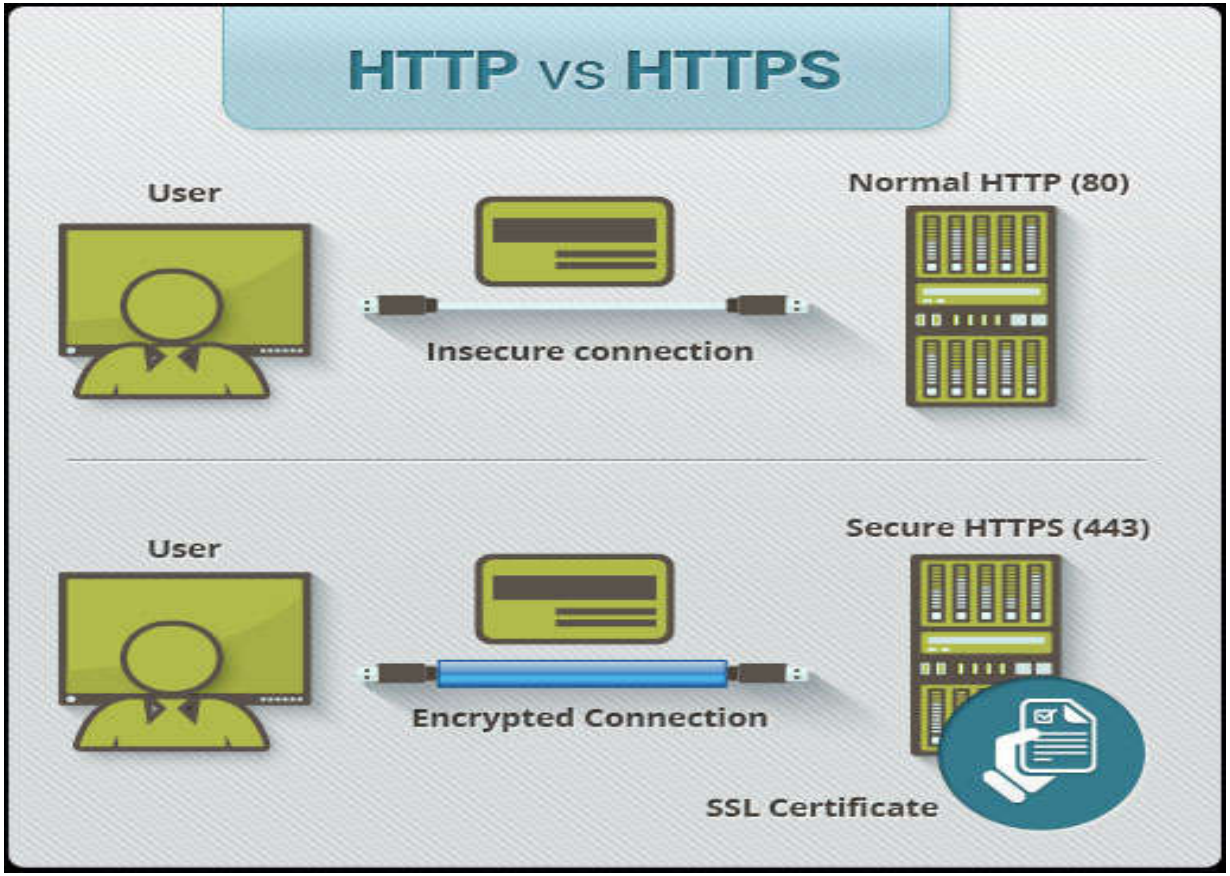
Tại sao phải bảo mật trong web

- Hoạt động kinh doanh bị gián đoạn, ngắt quãng
- Tránh đánh mất dữ liệu người dùng và lộ thông tin cá nhân
- Thứ hạng các từ khóa trên Google cũng bị mất ảnh hưởng đến quá trình SEO
- Uy tín hình ảnh thương hiệu cũng bị ảnh hưởng khi website bị tấn công
- Không thể sử dụng các loại hình quảng cáo trả phí như Facebook Ads, Google Ads…
- Thông qua những thông tin của doanh nghiệp, hacker có thể nắm được chiến lược kinh doanh của công ty.

HTTPS

- HTTPS (Hyper Text Transfer Protocol Secure - giao thức truyền tải siêu văn bản bảo mật) là phiên bản an toàn của HTTP, giao thức mà nhờ đó dữ liệu được gửi giữa trình duyệt và trang web bạn đang kết nối.
- Chữ 'S' ở cuối HTTPS là "Secure" (Bảo mật). Tất cả các giao tiếp giữa trình duyệt và trang web đều được mã hóa.
- HTTPS thường được sử dụng để bảo vệ các giao dịch trực tuyến có tính bảo mật cao như giao dịch ngân hàng và đặt hàng mua sắm trực tuyến.

HTTPS (2)


HTTP vs HTTPS
User
Normal HTTP (80)
Insecure connection
User
Secure HTTPS (443)
Encrypted Connection
SSL Certificate

HTTPS hoạt động như thế nào?

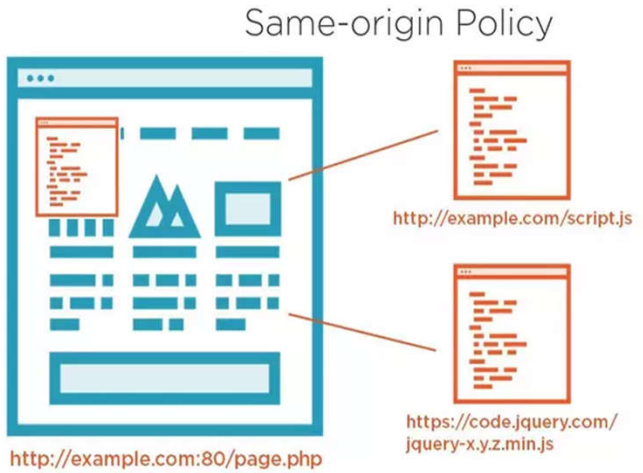
- HTTPS thường sử dụng một trong hai giao thức bảo mật để mã hóa thông tin liên lạc:
  - SSL (Secure Sockets Layer, tầng ổ bảo mật)
  - TLS (Transport Layer Security, bảo mật tầng truyền tải).
- Cả hai giao thức TLS và SSL đều sử dụng hệ thống PKI (Public Key Infrastructure, hạ tầng khóa công khai) bất đối xứng.
- Kết nối HTTPS với trang web, cần chứng chỉ SSL
- Chứng chỉ này chứa khóa công khai để bắt đầu phiên bảo mật, từ đó sẽ bắt đầu giao thức SSL handshake (giao thức bắt tay) để kết nối trang web.

Tại sao phải có chứng chỉ SSL

- Thông tin liên lạc gửi qua kết nối HTTP nằm trong văn bản thuần và có thể được đọc bởi bất kỳ hacker nào có thể đột nhập vào kết nối giữa trình duyệt và trang web của bạn.
- Với kết nối HTTPS, tất cả các thông tin liên lạc đều được mã hóa an toàn. Khi ai đó đã đột nhập vào kết nối, họ sẽ không thể giải mã bất kỳ dữ liệu nào đi qua giữa bạn và trang web.
- Lợi ích của giao thức HTTPS
  - Thông tin khách hàng, như số thẻ tín dụng, được mã hóa.
  - Khách truy cập có thể xác minh bạn là một doanh nghiệp đã đăng ký và bạn sở hữu tên miền.
  - Nhận được niềm tin của khách hàng.

Same-Origin Policy – SOP

- Mô hình bảo mật của trình duyệt web dựa trên chính sách cùng nguồn gốc (Same Origin Policy - SOP).
- Mô hình này cung cấp một vài đặc trưng bảo vệ cơ bản cho ứng dụng web.
- SOP là một trong những chính sách bảo mật quan trọng nhất trên trình duyệt hiện đại, nhằm ngăn chặn JavaScript code có thể tạo ra những request đến những nguồn khác với nguồn mà nó được trả về.



Same-Origin Policy – SOP (2)

- Ba tiêu chí chính để so sánh request bao gồm:
  - Domain (tên miền)
  - Protocol (giao thức)
  - Port (cổng kết nối)
- Nói đơn giản thì request sẽ được coi là hợp lệ chỉ khi nó thỏa mãn 3 tiêu chí ở trên (cùng domain, cùng protocol và cùng port)

Same-Origin Policy – SOP (3)

- Ví dụ: Chúng ta đang mở 2 tab, 1 tab là facebook, tab kia là 1 trang web nào đó có chứa mã độc. Sẽ rất nguy hiểm nếu như các đoạn script ở bên tab chứa mã độc có thể tự do thao tác lên tab facebook phía bên kia, và SOP sinh ra với nhiệm vụ ngăn chặn các hành động này.
- Dưới đây là ví dụ về list các pages vi phạm SOP của site origin là http://www.example.com:
  - http://www.example.co.uk (khác domain)
  - http://example.org (khác domain)
  - https://example.com (khác protocol)
  - http://example.com:8080 (khác port)

Bypass Same-Origin Policy

- Mặc dù ưu điểm bảo mật của SOP là rõ ràng, tuy nhiên trong một số trường hợp điều này lại gây khó khăn cho các nhà phát triển.
- Ví dụ Internet Explorer có hai ngoại lệ có thể bỏ qua SOP:
  - Nếu 2 domain cùng thuộc trust zone
  - Không bao gồm cổng, tức là http://company.com:81/index.html và http://company.com/index.html đều cùng source.
- Nếu một công ty có nhiều web application cùng yêu cầu xác thực tại một nơi, vd như http://store.company.com cần xác thực tại http://login.company.com trước. Việc nhận dữ liệu trả về từ request đến http://login.company.com không khả thi vì đã bị chặn do vi phạm SOP.

CORS

- CORS (Cross-origin resource sharing) là một cơ chế cho phép nhiều tài nguyên khác nhau (fonts, Javascript, v.v…) của một trang web có thể được truy vấn từ domain khác với domain của trang đó.
- CORS tích hợp trong HTML5, thêm vào các HTTP headers để sử dụng và quản lý nội dung cross-domain, cho phép lấy dữ liệu XMLHttpRequest.
- Lỗi thường gặp: **no 'access-control-allow-origin' header is present on the requested resource**
- Lỗi CORS policy: Khi call API mà không có header Access-Control-Allow-Origin hoặc giá trị không hợp lệ thì sẽ có lỗi này và không lấy được dữ liệu từ API.
- Cách khắc phục lỗi trên là phải config enable CORS lên để phía client có thể gọi được dữ liệu.

Tại sao lại cần CORS

- CORS được sinh ra là vì SOP, là một chính sách liên quan đến bảo mật được cài đặt vào toàn bộ các trình duyệt hiện nay.
- Chính sách này ngăn chặn việc truy cập tài nguyên của các domain khác một cách vô tội vạ.
- Ví dụ: Bạn truy cập một trang web có mã độc. Trang web đó sử dụng Javascript để truy cập tin nhắn Facebook của bạn ở địa chỉ https://facebook.com/messages.
  - Nếu bạn đã đăng nhập Facebook từ trước rồi. Nếu không có SOP, trang web độc hại kia có thể thoải mái lấy dữ liệu của bạn và bất cứ điều gì chúng muốn.
  - Same-origin policy chính là để ngăn chặn những kịch bản như trên để bảo vệ người dùng, giúp an toàn hơn khi lướt web.

CORS hoạt động ra sao

- Khi trình duyệt gửi một request đến một domain khác, những request này sẽ được gắn thêm một header có tên là origin để xác định origin của client, giá trị này được thêm tự động bởi trình duyệt và không ai có thể thay đổi nó được. Header này đại diện cho nguồn gốc truy vấn.
- Orgin được cấu tạo dựa trên ba phần:
  - Protocol/Scheme: (Http/Https)
  - Host: server/domain
  - Port: cổng, nếu giá trị này là giá trị mặc định 80 thì không cần
- Server kiểm tra origin trong request có hợp lệ hay không, nếu hợp lệ sẽ trả về response kèm header Access-Controll-Allow-Origin. Header này cho biết client có hợp lệ hay không rồi từ đó trình duyệt mới tiếp tục thực hiện quá trình request.

CORS hoạt động ra sao (2)

- Access-Control-Allow-Origin liệt kê các tên miền được phép thực hiện yêu cầu của CORS. Ký tự * cho phép tất cả các domain khác thực hiện yêu cầu. Cách này không an toàn ngoại trừ API của bạn được sử dụng với mục đích công khai và ai cũng có quyền được phép truy cập.
- Ví dụ một request từ trang https://foo.example

```
GET /resources/public-data/ HTTP/1.1
Host: bar.other
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Connection: keep-alive
Origin: https://foo.example
```

CORS hoạt động ra sao (3)

- Response từ hệ thống

  HTTP/1.1 200 OK

  Date: Mon, 01 Dec 2008 00:23:53 GMT

  Server: Apache/2

  Access-Control-Allow-Origin: *

  Keep-Alive: timeout=2, max=100

  Connection: Keep-Alive

  Transfer-Encoding: chunked

  Content-Type: application/xml

- Nếu không có header Access-Control-Allow-Origin hoặc giá trị của nó không hợp lệ thì trình duyệt sẽ thông báo lỗi.

Các header liên quan đến CORS

- Đều có phần tiền tố bắt đầu bằng Access-Controll. Ngoài Access-Controll-Allow-Origin, header liên quan đến CORS có thể chứa:
  - Access-Control-Allow-Methods: danh sách các phương thức HTTP mà server cho phép client sử dụng (GET, POST, PUT, DELETE...), không thể được ghi đè hay sửa đổi.
  - Access-Control-Allow-Headers: chứa danh sách những header mà phía server hiện đang hỗ trợ (x-authentication-token, ...), nếu trong request phía client gửi không chứa những header không nằm trong danh sách này sẽ bị server bỏ qua.
  - Access-Control-Max-Age: Mô tả thời gian hợp lệ của preflight request, nếu quá hạn, trình duyệt sẽ tạo môt pre-flight request mới.
- Pre-flight request là một loại CORS request, được gửi khi có những request tác động đến database như POST, PUT, DELETE,...

Các truy vấn dùng CORS

- Theo chuẩn quốc tế, các truy vấn sau sẽ phải dùng CORS:
  - Các truy vấn bằng XMLHttpRequest hoặc Fetch API đến một domain khác.
  - Ảnh, video được vẽ vào canvas sử dụng drawImage.
  - Web fonts truy vấn đến domain khác qua fontface trong CSS, trong đó trang web chỉ có thể sử dụng font dạng True Type nếu được cho phép.
  - WebGL Texture

Các truy vấn dùng CORS

- Theo chuẩn quốc tế, các truy vấn sau sẽ phải dùng CORS:
  - Các truy vấn bằng XMLHttpRequest hoặc Fetch API đến một domain khác.
  - Ảnh, video được vẽ vào canvas sử dụng drawImage.
  - Web fonts truy vấn đến domain khác qua fontface trong CSS, trong đó trang web chỉ có thể sử dụng font dạng True Type nếu được cho phép.
  - WebGL Texture

Web tracking

- Web tracking là cách mà người làm quản trị web thu nhập và chia sẽ những thông tin về hoạt động cụ thể cùng mỗi cá nhân người dùng trên World Wide Web.
- Với việc phân tích những hoạt động và thói quen của người sử dụng web của mỗi cá nhân sẽ đem lại cho các bạn những nội dung có liên quan đến sở thích cũng như nhu cầu của họ.

Web tracking (2)

- Một Web Tracker có nhiệm vụ vô cùng đơn giản đó chính là lưu trữ những hoạt động của bạn khi lên web.
- Chúng có thể lưu lại tên của bạn (nếu bạn có đăng kí account bất kỳ nào đó), hơn nữa chúng có thể biết được bạn thích gì, bạn đang xem theo dõi trang web nào nhiều trong thời gian gần đây.
- Có một vài web tracking không sử dụng hết nguồn tài nguyên thu thập. Do đó chúng thường chia sẻ những tài liệu đó với những trang web khác nhằm dựng lên một profile toàn diện hơn về bạn. Qua đó các bạn sẽ có được những gợi ý để quảng cáo đến người có nhu cầu.
- Có nhiều bạn thắc mắc khi sử dụng facebook "tại sao lại thấy những mẩu quảng cáo về những món mình thích trên newfeed?"

Bạn bị theo dõi bởi web tracking ra sao?

- Những đoạn code các bạn vẫn lầm tưởng là bugs thực tế lại không phải. Bởi lẽ chúng được nhúng trên web và thông báo tới các nhà Quản trị trang web mỗi khi mà bạn vào thăm một trang page bất kỳ.
- Nhiều Web sử dụng cookies để thu thập những thông tin trong quá khứ sử dụng web của bạn. Đặc biệt là mạng xã hội luôn thu thập thông tin của bạn để quảng cáo của họ trở nên hiệu quả hơn.
- Tất cả các hoạt động của bạn đều được theo dõi mà bạn không hề hay biết thậm chí là không có sự đồng ý của bạn.
- Gần như tất cả các web mà bạn thích, họ đều theo dõi phiên làm việc của bạn. Bởi lẽ họ cần những thông tin trong quá khứ của bạn để định hướng quảng cáo trong tương lai trở nên chính xác hơn.

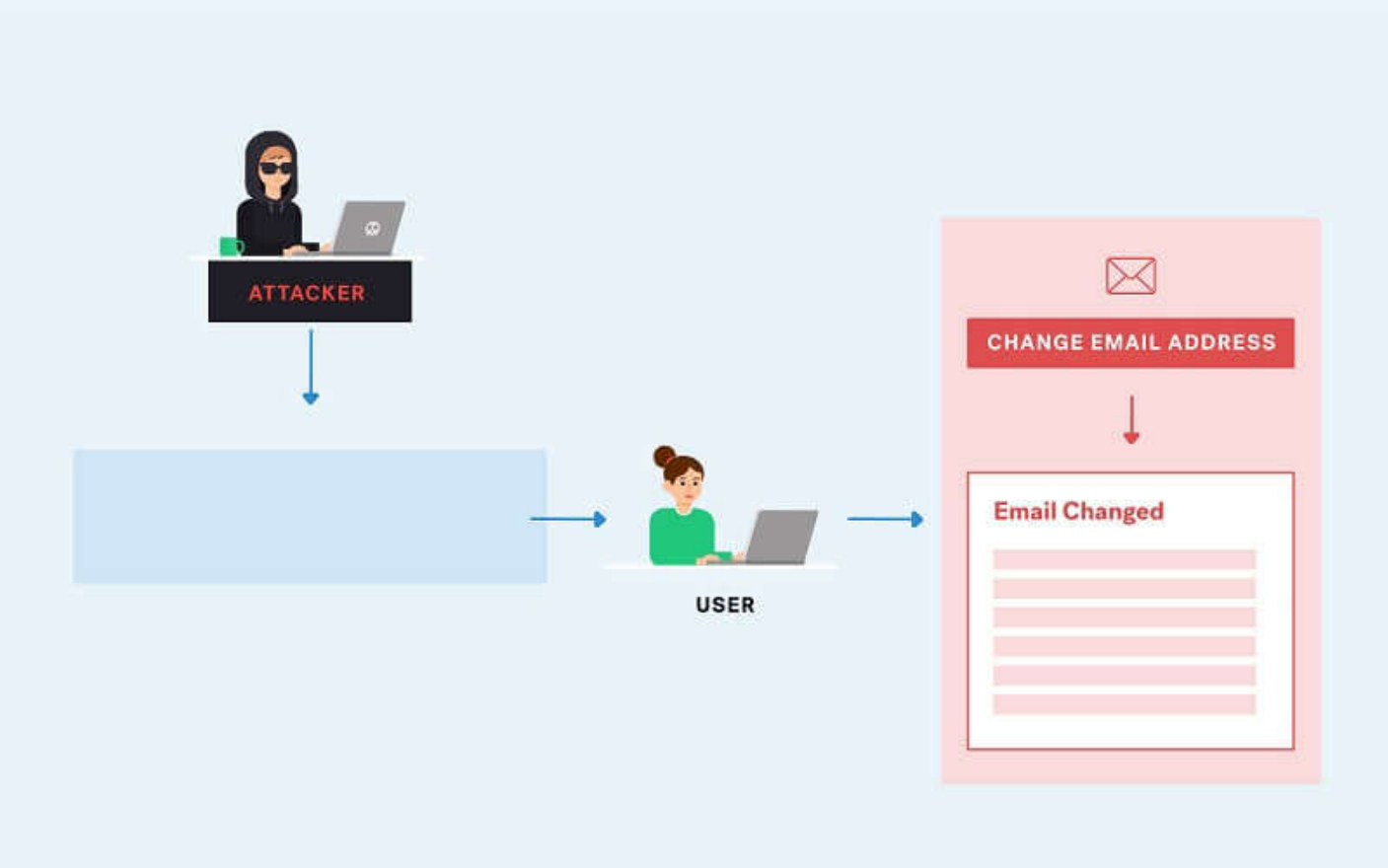
Một số kĩ thuật Tracking

- **Web Beacon**: là những object nhỏ và được nhúng trực tiếp vào những trang web, đồng thời vào email và được kích hoạt bất kỳ khi nào bạn kích hoạt trang web hoặc email có chứa chúng
- **Flash Cookies:** hay còn có tên khác là Local Shared Object.
  - Tồn tại dưới dạng một file text được gửi từ web server tới người dùng khi browser request tới những nội dung cần đến Adobe Flash.
  - Những thông tin được lưu bởi Flash Cookie sẽ nằm ở một file Adobe và được quản lý bởi Adobe Flash Player
- **Canvas fingerprinting:** Thẻ canvas HTML5 dùng để vẽ đồ họa trên web thông qua việc sử dụng các script của JavaScript. Ngoài ra nó còn được dùng để lấy Browser Fingerprinting để làm tracking online.

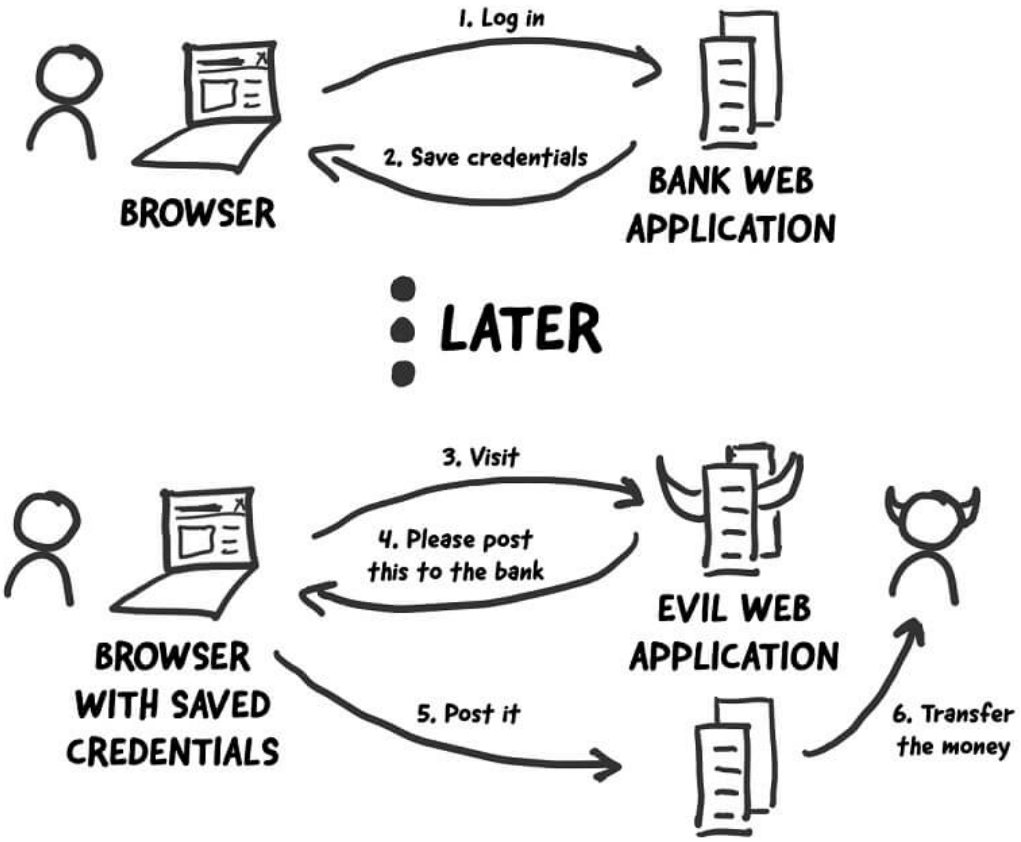
Cross-Site Request Forgery (CSRF)

- CSRF là **Kỹ thuật tấn công giả mạo chính chủ thể của nó**. Tức là lạm dụng sự tin tưởng ứng dụng web trên trình duyệt của nạn nhân.
- CSRF nói đến việc tấn công vào chứng thực request trên web thông qua việc sử dụng Cookies. Đây là nơi mà các hacker có khả năng sử dụng thủ thuật để tạo request mà bạn không hề biết.
- Các kiểu tấn công CSRF xuất hiện từ những năm 1990, đến năm 2007 mới có một vài tài liệu miêu tả chi tiết về các trường hợp tấn công CSRF.
- Năm 2008 có khoảng 18 triệu người sử dụng eBay ở Hàn Quốc mất các thông tin cá nhân của mình. Cũng trong năm đó, một số khách hàng tại ngân hàng Mexico bị mất tài khoản cá nhân của mình. Trong 2 trường hợp trên hacker đều sử dụng kĩ thuật tấn công CSRF.

Cross-Site Request Forgery (CSRF)



Cách thức hoạt động của CSRF



Cách ngăn chặn CSRF từ phía client

- Sau khi kết thúc một phiên làm việc, nên thực hiện đăng xuất ra khỏi các website, đặc biệt là những website quan trọng.
- Nên đăng nhập vào một thiết bị chuyên biệt và không cho ai sử dụng thiết bị đó
- không nên nhấp vào các liên kết mà bạn nhận được qua mạng xã hội, thư điện tử,...
- Không nên save lại các dữ liệu về password tại trình duyệt của mình. Không nên chọn các cách thức "lưu mật khẩu", "nhớ mật khẩu" hay "click một lần để đăng nhập",...
- Không nên truy cập vào những website khác trong quá trình thực hiện các giao dịch hay đang tương tác trên những website quan trọng. Điều này có thể chứa đựng những mã thu thập của hacker.

Cách ngăn chặn CSRF từ hệ thống

- Sử dụng các thông báo xác thực, captcha, OTP, …
- Dùng csrf_token: Việc sử dụng token sẽ làm biến đổi thường xuyên trong phiên làm việc. Và khi thông tin được biến đổi đính kèm với dữ liệu token này. Nếu chúng được sinh ra không trùng với token được gửi lên thì loại bỏ ngay yêu cầu lên hệ thống.
- Dùng tập tin chuyên biệt cho trang quản trị: Nên để trang admin chuyên biệt ở một subdomain, điều này làm chúng không dùng chung một tập tin với lập trình giao diện của sản phẩm.
- Kiểm tra IP: Đối với các hệ thống quan trọng, chỉ nên cho phép truy cập vào hệ thống với những địa chỉ IP được xác lập riêng và có sẵn, hoặc chỉ trao quyền admin cho IP VPN hoặc local mà thôi.

Tấn công Cross-Site Scripting (XSS)

- là kỹ thuật tấn công bằng cách chèn vào các website những đoạn mã script nguy hiểm như javascript hoặc HTML. Thông thường, các cuộc tấn công XSS được thực thi ở phía client, vượt qua quyền kiểm soát truy cập, chiếm phiên đăng nhập và mạo danh người dùng.
- XSS là các request được gửi từ máy client tới server nhằm chèn vào đó các thông tin vượt quá tầm kiểm soát của server. Nó có thể là request được gửi từ các form dữ liệu đầu vào hoặc từ các URL.
- XSS chỉ gây tổn hại đối với website ở phía client nên sẽ gây hậu quả trực tiếp cho người dùng. XSS không làm ảnh hưởng đến hệ thống website nằm trên server.
- Mục tiêu tấn công của XSS là người dùng website khi vô tình vào các trang có chứa mã độc nguy hiểm từ đó bị mất thông tin người dùng, bị thay đổi thiết lập, …

Các kiểu khai thác XSS

- XSS được chia làm 3 loại chính là Reflected, Stored và DOM based.
  - 75% kỹ thuật XSS dựa trên Reflected XSS. Hacker gửi cho người dùng một URL có chứa đoạn mã nguy hiểm, chỉ cần click vào là người dùng sẽ gặp nguy hiểm.
  - Stored XSS hướng đến nhiều nạn nhân hơn khi ứng dụng web không kiểm tra kỹ các dữ liệu đầu vào trước khi lưu vào database. Hacker sẽ thông qua các điểm đầu vào (form, textbox, textarea…) không được kiểm tra kỹ để chèn vào CSDL các đoạn mã nguy hiểm.
  - DOM Based XSS là kỹ thuật khai thác XSS dựa trên việc thay đổi cấu trúc DOM của tài liệu, cụ thể là HTML.

SQL Injection là gì?

- SQL Injection là kỹ thuật tấn công vào CSDL của ứng dụng thông qua việc khai thác các lỗ hổng bảo mật nhằm mục đích thay đổi hoặc thậm chí phá hủy hệ thống CSDL
- SQL Injection thường được thực hiện thông qua form nhập liệu để chèn vào câu truy vấn SQL

SQL Injection là gì?

- Cho form như sau: hãy tìm kiếm trong bảng students nhân viên có đúng hoặc gần đúng dựa vào name mà user nhập

Nhập tên: Tìm kiếm

- Nhập chuỗi sau vào ô tìm kiếm: nothing' or name <> ', kết quả là toàn bộ các bản ghi trong bảng students được show ra!

Cách phòng chống SQL Injection

- Luôn validate chặt chẽ dữ liệu từ client
- Với code thuần, luôn sử dụng hàm mysqli_real_escape_string($connection, $query) để lọc dữ liệu từ client
- Sử dụng parameter thay vì cộng chuỗi khi tạo câu truy vấn
- Không hiển thị exception, message thông báo lỗi
- Phân quyền rõ ràng trong Database
- Backup dữ liệu thường xuyên