SQL injection

What is SQLi?

- SQL injection là một lỗ hổng bảo mật web cho phép kẻ tấn công can thiệp vào các truy vấn mà một ứng dụng thực hiện vào cơ sở dữ liệu của nó. Nó thường cho phép kẻ tấn công xem dữ liệu mà họ thường không thể truy xuất. Điều này có thể bao gồm dữ liệu thuộc về những người dùng khác hoặc bất kỳ dữ liệu nào khác mà chính ứng dụng có thể truy cập. Trong nhiều trường hợp, kẻ tấn công có thể sửa đổi hoặc xóa dữ liệu này, gây ra những thay đổi liên tục đối với nội dung hoặc hành vi của ứng dụng.
- Trong một số trường hợp, kẻ tấn công có thể leo thang tấn công sql tiêm để thỏa hiệp máy chủ cơ bản hoặc cơ sở hạ tầng back-end khác hoặc thực hiện một cuộc tấn công từ chối dịch vụ.

How does SQLi work?

Có nhiều kiểu tấn công bằng SQL Injection tuỳ thuộc vào Database Engine:

- **SQL statement that is always true:** Tin tặc thực hiện tiêm SQL với câu lệnh SQL luôn đúng.

  Ví dụ: 1=1; thay vì chỉ nhập đầu vào "sai", tin tặc sử dụng một tuyên bố sẽ luôn đúng.

  Nhập "100 OR 1=1" vào hộp nhập truy vấn sẽ trả về phản hồi với các chi tiết của bảng.
- **"OR ""=":** Kẻ tấn công cần nhập "OR ""=" vào hộp nhập truy vấn. Hai dấu hiệu này đóng vai trò là mã độc để đột nhập vào ứng dụng.

Ví dụ: Kẻ tấn công tìm cách truy xuất dữ liệu người dùng từ một ứng dụng và chỉ cần nhập "OR =" trong ID người dùng hoặc mật khẩu. Vì câu lệnh SQL này là hợp lệ và đúng, nó sẽ trả về dữ liệu của bảng người dùng trong cơ sở dữ liệu.

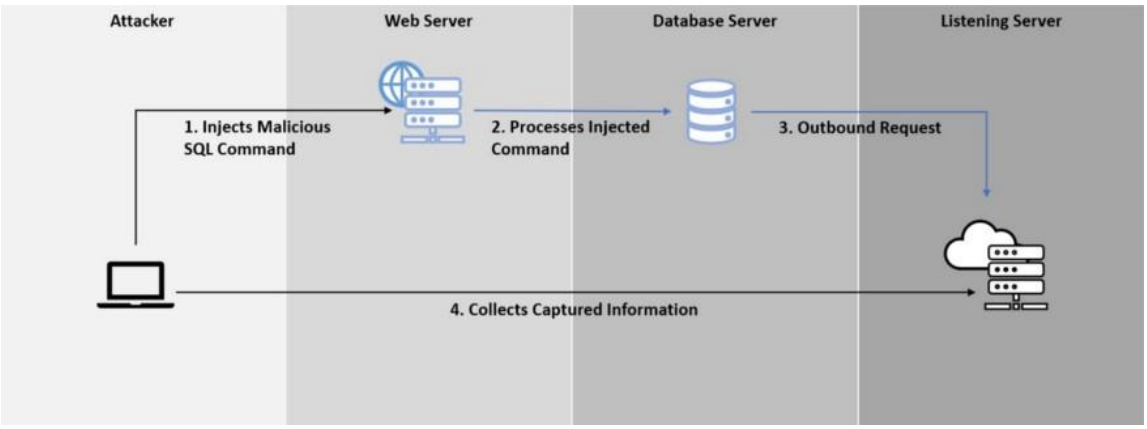
- **Batched SQL injection:** Tiêm SQL theo lô bao gồm một tập hợp các câu lệnh SQL được phân tách bằng dấu chấm phẩy. Điều duy nhất có thể làm cho cách tiếp cận này thành công là nếu các câu lệnh SQL là đúng và hợp lệ— nghĩa là câu lệnh sau dấu chấm phẩy cần phải đúng.

Ví dụ: nếu tin tặc nhập "105; DROP TABLE Supplier," câu lệnh SQL sau dấu chấm phẩy sẽ xóa bảng nhà cung cấp khỏi cơ sở dữ liệu ứng dụng.

How many types of the SQLi?

SQLi được chia làm 3 loại: In-band SQLi, Inferential (Blind) SQLi, Out-of-band SQLi

- In-band SQLi: Kẻ tấn công sử dụng cùng một kênh liên lạc để khởi động các cuộc tấn công của họ và thu thập kết quả của họ. Sự đơn giản và hiệu quả của In-band SQLi làm cho nó trở thành một trong những loại tấn công SQLi phổ biến nhất. Có hai biến thể phụ của phương pháp này:
    - **Error-based SQLi**: Kẻ tấn công thực hiện các hành động khiến cơ sở dữ liệu tạo ra thông báo lỗi. Kẻ tấn công có khả năng sử dụng dữ liệu được cung cấp bởi các thông báo lỗi này để thu thập thông tin về cấu trúc của cơ sở dữ liệu.
    - **Union-based SQLi:** Kỹ thuật này tận dụng toán tử UNION SQL, hợp nhất nhiều câu lệnh được chọn được tạo bởi cơ sở dữ liệu để có được một phản hồi HTTP duy nhất. Phản hồi này có thể chứa dữ liệu có thể được tận dụng bởi những kẻ tấn công.
- **Inferential (Blind) SQLi:** Kẻ tấn công gửi tải trọng dữ liệu đến máy chủ và quan sát phản hồi và hành vi của máy chủ để tìm hiểu thêm về cấu trúc của nó. Phương pháp này được gọi là SQLi mù vì dữ liệu không được chuyển từ cơ sở dữ liệu trang web cho kẻ tấn công, do đó kẻ tấn công không thể xem thông tin về cuộc tấn công trong băng tần. Tiêm SQL mù dựa vào phản ứng và mô hình hành vi của máy chủ vì vậy chúng thường chậm hơn để thực hiện nhưng có thể cũng có hại. Tiêm SQL mù có thể được phân loại như sau:
    - **Boolean:** Kẻ tấn công gửi truy vấn SQL đến cơ sở dữ liệu nhắc ứng dụng trả về kết quả. Kết quả sẽ khác nhau tùy thuộc vào việc truy vấn là đúng hay sai. Dựa trên kết quả, thông tin trong phản hồi HTTP sẽ sửa đổi hoặc không thay đổi. Kẻ tấn công sau đó có thể làm việc nếu thư tạo ra kết quả đúng hoặc sai.
    - **Time-based**: Kẻ tấn công gửi truy vấn SQL đến cơ sở dữ liệu, khiến cơ sở dữ liệu chờ (trong một khoảng thời gian tính bằng giây) trước khi nó có thể phản ứng. Kẻ tấn công có thể xem từ khi cơ sở dữ liệu mất để đáp ứng, cho dù truy vấn là đúng hay sai. Dựa trên kết quả, phản hồi HTTP sẽ được tạo ngay lập tức hoặc sau một khoảng thời gian chờ đợi. Do đó, kẻ tấn công có thể làm việc nếu thông báo họ sử dụng trả về đúng hoặc sai, mà không cần dựa vào dữ liệu từ cơ sở dữ liệu.
- **Out-of-band SQLi:**



So với In-Band và Blind SQL Injection, OOB SQL injection lọc dữ liệu thông qua kênh đi, có thể là giao thức DNS hoặc HTTP. Khả năng của một hệ thống cơ sở dữ liệu để bắt đầu yêu cầu DNS hoặc HTTP bên ngoài có thể cần phải dựa vào chức năng có sẵn. Hàm có thể là hàm hoạt động của tệp (ví dụ: load_file(), master.. xp_dirtree) hoặc thiết lập chức năng kết nối (ví dụ: DBMS_LDAP. INIT, UTL_HTTP.request). Để khai thác OOB SQL injection, các máy chủ web và cơ sở dữ liệu được nhắm mục tiêu phải đáp ứng các điều kiện sau:

- **T**hiếu xác thực đầu vào trên ứng dụng web
- Môi trường mạng cho phép máy chủ cơ sở dữ liệu được nhắm mục tiêu bắt đầu yêu cầu gửi đi (DNS hoặc HTTP) đến nơi công cộng mà không bị hạn chế chu vi bảo mật
- Đủ đặc quyền để thực hiện hàm cần thiết để bắt đầu truy vấn lại bên ngoài

Impact of SQLi:

- Một cuộc tấn công tiêm SQL thành công có thể dẫn đến truy cập trái phép vào dữ liệu nhạy cảm, chẳng hạn như mật khẩu, chi tiết thẻ tín dụng hoặc thông tin người dùng cá nhân. Nhiều vi phạm dữ liệu cao cấp trong những năm gần đây là kết quả của các cuộc tấn công tiêm SQL, dẫn đến thiệt hại danh tiếng và tiền phạt theo quy định. Trong một số trường hợp, kẻ tấn công có thể có được một cửa hậu liên tục vào hệ thống của một tổ chức, dẫn đến một thỏa hiệp lâu dài có thể không được chú ý trong một thời gian dài.

How to prevent SQLi attack?

- Xác thực đầu vào theo các cách;

    - Phía máy khách VÀ quan trọng nhất là xác thực phía máy chủ thông qua việc sử dụng danh sách trắng
    - Sử dụng Câu lệnh SQL đã chuẩn bị với các truy vấn được tham số hóa
    - Sử dụng quy trình SQL được lưu trữ
    - Escape (một loại mã hóa) tất cả các đầu vào do người dùng cung cấp.

Reference:

- https://www.ptsecurity.com/ww-en/analytics/knowledge-base/how-to-prevent-sql-injection-attacks/
- https://www.imperva.com/learn/application-security/sql-injection-sqli/
- https://portswigger.net/web-security/sql-injection
- https://portswigger.net/web-security/sql-injection
- https://viblo.asia/p/sql-injection-va-cach-phong-chong-OeVKB410lkW