1. **SQL injection**: - Là kỹ thuật sử dụng câu lệnh SQL chèn vào phần input của website, làm thay đổi kết quả trả về của câu truy vấn

Qua đó, attacker có thể lấy được các thông tin quan trọng từ website: username, pw trang admin, tài khoản ngân hàng.

* **Các tấn công:**

**Dạng tấn công vượt qua kiểm tra lúc đăng nhập**

- Với dạng tấn công này, attacker có thể dễ dàng vượt qua các trang đăng nhập nhờ vào lỗi khi dùng các câu lệnh SQL thao tác trên cơ sở dữ liệu của ứng dụng web.

**Dạng tấn công khi bị rò rỉ thông tin do sử dụng câu lệnh UNION SELECT**

- Để thực hiện được kiểu tấn công này, attacker phải có khả năng hiểu và lợi dụng các sơ hở trong các thông báo lỗi từ hệ thống để dò tìm các điểm yếu khởi đầu cho việc tấn công.

**Dạng tấn công sử dụng câu lệnh INSERT**

* Thông thường các ứng dụng web cho phép người dùng đăng ký một tài khoản để tham gia. Chức năng không thể thiếu là sau khi đăng ký thành công, người dùng có thể xem và hiệu chỉnh thông tin của mình. SQL injection có thể được dùng khi hệ thống không kiểm tra tính hợp lệ của thông tin nhập vào.

**Dạng tấn công sử dụng stored-procedures**

* Stored Procedure được sử dụng trong lập trình web với mục đích nhằm giảm sự phức tạp trong ứng dụng và tránh sự tấn công trong kỹ thuật sql injection.Tuy nhiên những kẻ tấn công vấn có thể lợi dụng những stored procedure để tấn công vào hệ thống.Việc tấn công này sẽ gây tác hại rất lớn nếu ứng dụng được thực thi với quyền quản trị hệ thống ‘sa’.

**Dạng tấn công sử dụng Blind SQL Injection**

* Một số ứng dụng Web đã được thiết kế xử lý lỗi khá tốt và về cơ bản kẻ tấn công không thể thấy được các thông báo lỗi chứa các thông tin nhạy cảm. Do vậy việc tấn công bằng cách sử dụng từ khoá UNION hay tấn công dựa vào thông báo lỗi không thể thực hiện. Trong trường hợp này thì kẻ tấn công sẽ sử dụng kỹ thuật Blind SQL Injection. Có hai dạng Blind SQL Injection:

**Normal Blind:** Không thấy phản hồi từ trang Web nhưng kẻ tấn công có thể thấy được kết quả thông qua mã trạng thái HTTP hoặc kết quả của câu truy vấn. Cách tấn công Normal Blind thường được thực hiện bằng cách sử dụng câu điều kiện If hoặc tấn công vào mệnh đề Where

**Totally Blind:** Hoàn toàn không thấy bất cứ phản hồi nào từ trang Web. Khi đó kẻ tấn công sẽ sử dụng một số hàm hệ thống và phân tích sự trả lời từ trang Web để tìm ra lỗ hổng.

**Dạng tấn công khi bị rò rỉ thông tin thông qua error – message**

* Nó là phần nội dung có thể dùng để khai thác các thông tin liên quan đến hệ thống, các thông tin từ folder list, thông tin kết nối database, thông tin user đến thông tin password …
* **Cách phòng chống:**
  + Nếu DL là kiểu INT, nên ép kiểu, chuyển về kiểu INT, sau đó chuyển về kiểu string
  + Nếu DL là kiểu chuỗi, thiết lập độ dài tối đa cần thiết, lọc bỏ ký tự nguy hiểm, lọc bỏ các từ khóa
  + Viết lại đường dẫn có thể chống SQL injection
  + Phân quyền CSDL
  + Sử dụng placeholder
  + Ngăn hiển thị error – message

1. **Cross Site Request Forgery (CSRF):** là kỹ thuật tấn công bằng cách sử dụng quyền chứng thực của người dùng đối với một website.

* **Phương thức tấn công**: Tấn công theo kiểu input-run: Hacker sử dụng phương pháp CSRF để lừa trình duyệt của người dùng gửi đi các câu lệnh http đến các ứng dụng web. Điều đó có thể thực hiện bằng cách chèn mã độc hay link đến trang web mà người dùng đã được chứng thực. Trong trường hợp phiên làm việc của người dùng chưa hết hiệu lực thì các câu lệnh trên sẽ được thực hiện với quyền chứng thực của người sử dụng
* **Cách phòng chống**:
  + **Phía user:**

- Nên thoát khỏi các website quan trọng

- Không nên click vào các đường dẫn mà nhận được qua email, qua fb

- Không lưu các thông tin về mật khẩu tại trình duyệt của mình

- Trong quá trình thực hiện giao dịch hay vào các website quan trọng không nên vào các website khác, có thể chứa mã khai thác của kẻ tấn công

- Input lại password

* + **Phía server:**

- lựa chọn việc sử dụng GET và POST đúng cách

- Sử dụng captcha, các thông báo xác nhận

- Sử dụng token

- Sử dụng cookie riêng biệt cho trang quản trị

- Kiểm tra REFERRER

- Kiểm tra IP

1. **Cross Site Scripting (XSS):** là lỗ hổng cho phép hacker có thể chèn những đoạn mã client – script (thường là javascript hoặc HTML) vào trang web, khi người dùng vào những trang này, mã độc sẽ được thực thi trên máy của người dùng

Lợi dụng lỗi XSS, hacker có thể lừa đảo quản trị viên của web, đánh cắp cookie, chiếm session … từ đó có thể đăng nhập chiếm quyền điều khiển website

* **Kỹ thuật tấn công**:
  + **Tấn công Persitent XSS (Stored XSS)**:
    - là dạng tấn công mà hacker chèn trực tiếp các mã độc vào CSDL của web
    - xảy ra khi các DL được gửi lên server không được kiểm tra kỹ mà lưu trực tiếp vào CSDL
* **Tấn công Non- Persitent XSS (Reflected XSS)**
* là dạng tấn công mà hacker không gửi DL độc hại lên server, mà gửi trực tiếp link có chứa mã độc cho người dùng, khi người dùng click vào link này thì trang web sẽ được load chung với các đoạn script độc hại
* thường dùng để ăn cắp cookie, chiếm session, cài keylogger, trojan, … vào máy tính của người dùng
* **Tấn công DOM-based XSS**:
  + - Là dạng tấn công XSS làm thay đổi cấu trúc của trang web bằng cách thay đổi cấu trúc HTML
    - Hacker có thể chèn các đoạn script nhằm làm thay đổi giao diện mặc định của trang web thành giao diện giả
* **Cách phòng chống**:
  + **Stored XSS:** mã hóa các ký tự <, > để vô hiệu hóa script trước khi insert nó vào DB
  + **Reflected XSS:** - trước khi in ra 1 từ nào đó ra màn hình thì mã hóa các ký tự <, >
* Trong XSS, thì nên để các class hoặc id trước khi thuộc tính value
* **Dom-based XSS**: - Sử dụng bộ lọc chuỗi giá trị đầu ra để loại bỏ các ký tự nhạy cảm
* Sử dụng 1 số công cụ khác: Dynamic Data Tainting, Static Analysis

1. **Send mail**

* **Lỗ hổng:** 
  + Lỗ hổng mail header injection:
  + Lưu giữ địa chỉ gửi đi dựa trên tham số hidden
  + Gửi qua người trung gian thứ 3 dựa vào mail server
* **Phòng chống:**
  + Không cho phép bao gồm tham số từ bên ngoài
  + Check khi gửi mail không bao gồm xuống hàng ở tham số bên ngoài
  + Sử dụng library hay API chuyên dụng để gửi mail
  + Check mail address
  + Check tiêu đề mail

1. **Session Hijacking:** là kỹ thuật đánh cắp session lợi dụng các lỗ hổng trong cơ chế tạo và quản lý session và các lỗ hổng hệ thống như XSS nhằm truy cập bất hợp pháp, đánh cắp thông tin người dùng

* **Cách tấn công:**
  + **Phỏng đoán session ID**
  + **Đánh cắp session ID**
  + **Cưỡng chế session ID**
* **Các lỗ hổng:**
  + XSS
  + HTTP header – injection
  + Session ID được chèn vào URL
  + Tự tạo ra session ID có nguy cơ cao gây mất an toàn
* **Cách phòng chống:**
  + Dùng bộ phát sinh số ngẫu nhiên an toàn
  + Không tự tạo session ID
  + Lưu session ID trên Cookie
  + Không lưu session trên URL
  + Fix các lỗi XSS, HTTP Header injection
  + Fix các lỗi về cookie
  + Thay đổi session ID sau khi chứng thực
  + Trước khi chứng thực không lưu thông tin bảo mật lên session