**1 . SQL Injection**

**Khái Quát**

SQL Injection là lỗ hổng phát sinh trong trường hợp có thiếu sót khi gọi lệnh SQL. Trong trường hợp có lỗ hổng SQL Injection trong ứng dụng thì có khả năng sẽ chịu những ảnh hưởng như dưới đây. Và trong tất cả trường hợp kẻ tấn công đều có thể chủ động tấn công vào server (không liên quan đến người sử dụng).

* Người ngoài có thể lấy toàn bộ thông tin trong CSDL.
* Có thể thay đổi nội dung trong CSDL.
* Tránh né việc xác nhận ( Đăng nhập mà không cần sử dụng ID và PASSWORD).
* Ngoài ra, có thể thực hiện những hành động như đọc và viết thêm vào file trên sever CSDL, chạy chương trình.

Đối sách xác thực của lỗ hổng SQL Injection là sử dụng placeholder và gọi lệnh SQL

**Cách tấn công**

* **Rò rỉ thông tin thông qua các Error Messenger**
* **Rò rỉ thông tin do sử dụng UNION SELECT ( ?author=’ union select id,pwd,name, addr,null, null from users -- )**
* **Tránh xác nhận bằng SQL Injection ( đăng nhập ko cần pass)**
* **Tạo dữ liệu giả bằng tấn công SQL Injection ( update users set name=’abc’ where id = 1 -- )**

**Phòng thủ:**

* **Place holder tĩnh, place holder động**
* **Đối sách bảo đảm của SQL Injection**
  + Ngăn các error message chi tiết . ( display\_errors = Off )
  + Kiểm tra tính hợp lý của giá trị input
  + Thiết lập quyền hạn của Database

2 CSRF là gì?

* CSRF ( Cross Site Request Forgery) là kỹ thuật tấn công bằng cách sử dụng quyền chứng thực của người dùng đối với một website. CSRF là kỹ thuật tấn công vào người dùng, dựa vào đó hacker có thể thực thi những thao tác phải yêu cầu sự chứng thực. Hiểu một cách nôm na, đây là kỹ thuật tấn công dựa vào mượn quyền trái phép.

Dưới đây là ví dụ ảnh hưởng khi có phát sinh các lỗ hổng CSRF trong ứng dụng Web

* Mua sản phẩm bởi account của người sử dụng
* Thực hiện xử lý huỷ dịch vụ của của người sử dụng
* Dùng account của người sử dụng để post bài lên trang thông báo
* Thay đổi password và địa chỉ email của người sử dụng

**Kỹ thuật tấn công và ảnh hưởng**

* **Kỹ thuật tấn công CSRF theo kiểu input-run**
* **Kỹ thuật tấn công CSRF trong trường hợp có màn hình xác nhận**
* **truyền tham số dựa vào biến số session**

**Phòng chống:**

* **Phân biệt những trang cần thiết phải phòng chống CSRF**
* **Xác nhận có phải request mà người dùng chính thức đặt ra hay không**
  + **Nhúng thông tin bí mật (token)**
  + **Kiểm tra Referer**
  + **Input lại password**

**3 Session**

* **Session Hijacking**
  + Phỏng đoán session ID
  + Đánh cắp session ID
  + Cưỡng chế session ID
* **ID session có thể phỏng đoán được**

Kỹ thuật tấn công đối với session ID có khả năng đoán được tiến hành bằng 3 bước dưới đây.

* Tập hợp các session ID từ ứng dụng thuộc đối tượng tấn công
* Xây dựng giả thuyết về tính quy tắc của session ID
* Thử các session ID đã đoán được trên ứng dụng đối tượng

**Phòng chống**

**Phương pháp cải thiện tính chất ngẫu nhiên (Randomness) của session ID PHP**

* Địa chỉ remote IP
* Thời gian hiện tại
* Random number (không phải là bộ phát sinh số giả ngẫu nhiên an toàn)
* **Session ID trong việc chèn vào URL**

Điều kiện để session ID bị rò rỉ bởi Referer là website thỏa mãn 2 điểm như dưới đây.

* Có thể sử dụng session ID chèn vào URL
* Có liên kết với website bên ngoài. Hoặc là người dùng có thể tạo đường link đó

**Biện pháp phòng chống**

Để không sử dụng session ID chèn URL, chúng ta sẽ thiết lập để lưu session ID trên Cookie.

Session.use\_cookies = 1

Session.use\_only\_cookies = 1

* **Cố định hóa session ID**
* Lấy session ID
* Thực hiện cưỡng chế session ID đã lấy được ở bước 1 đối với người bị hại
* Người bị hại sẽ login vào ứng dụng thuộc mục tiêu tấn công
* Kẻ tấn công sẽ sử dụng session ID đã cưỡng chế từ người bị hại để truy cập vào ứng dụng thuộc mục tiêu tấn công