1. **Injection**
2. **Injection là gì ?**

Injection là kỹ thuật tấn công mà kẻ tấn công sẽ tiêm những đoạn mã độc vào dữ liệu được gửi tới ứng dụng nhằm mục tiêu có thể thực hiện được những công viêc vượt quá quyền hạn vốn có của kẻ tấn công.

Trong Software Engineering thì có các khái niệm như SQL Injection, OS command Injection, Dependency Injection.

1. **SQL Injection**
   1. **SQL Injection là gì ?**

SQL Injection là kỹ thuật tấn công mà kẻ tấn công sẽ lợi dụng lỗ hổng trong việc kiểm tra dữ liệu đầu vào của ứng dụng và các thông báo lỗi của DBMS, nhằm tiêm những câu lệnh SQL vào dữ liệu được gửi tới ứng dụng. Từ đó có thể thực thi được những câu lệnh SQL mà không được cho phép trên hệ thống cơ sở dữ liệu của ứng dụng.

* 1. **Mức độ nghiêm trọng**

SQL Injection có thể cho phép kẻ tấn công vượt qua hệ thống xác thực, thực hiện các thao tác trên cơ sở dữ liệu của ứng dụng hay thậm chí là cả các thao tác trên server mà ứng dụng đang chạy.

* 1. **Phân loại SQL Injection**
     + **In-band SQLi**

Xảy ra khi kẻ tấn công có thể tổ chức tấn công và thu thập kết quả trực tiếp trên cùng một kênh liên lạc

In-band SQLi được chia làm 2 loại chính:

* + - * 1. **Error-based**

Là kĩ thuật tấn công dựa vào thông báo lỗi được trả về từ Database Server nhằm lấy được thông tin về cơ sở dữ liệu

VD: **Khai thác SQL injection bằng cách lợi dụng Numberic overflow trong MySQL**

MySQL hỗ trợ các kiểu dữ liệu tiêu chuẩn SQL như INTEGER, SMALLINT, TINYINT, MEDIUMINT, BIGINT. FLOAT, DOUBLE…

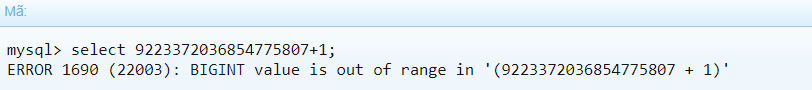
Mỗi kiểu dữ liệu sẽ có một phạm vi giới hạn khác nhau. Khi các giá trị vượt ra ngoài giới hạn này nó sẽ dẫn tới hiện tượng tràn số (overflow). Có sự khác nhau trong xử lý Out-of-range và overflow giữa các phiên bản MySQL. Việc xử lý Out-of-range và overflow từ phiên bản MySQL 5.5.5 trở lên sẽ trả về một thông báo lỗi

Lợi dụng xử lý out-of-range và overflow trên các phiên bản MySQL từ 5.5.5 trở lên, ta có thể gây ra tràn số để leak dữ liệu qua các thông báo lỗi, qua đó thực hiện khai thác Error based SQL injection.

Ví dụ, kiểu dữ liệu BIGINT có kích thước 8 byte (64 bit). Giá trị số nguyên có dấu lớn nhất của nó là:

* Binary: 0b0111111111111111111111111111111111111111111111111111111111111111
* Hex: 0x7fffffffffffffff
* Decimal: 9223372036854775807

MySQL 5.5.5 trở lên: Khi thực hiện phép toán với các giá trị trên (ví dụ tăng thêm 1) sẽ gây ra tràn số và trả về thông báo lỗi:



Hackers có thể dựa vào thông báo lỗi đó để tấn công.

* + - * 1. **Union-based**

Là kỹ thuật tấn công dựa vào toán tử UNION trong SQL cho phép tổng hợp kết quả của 2 hay nhiều câu SELECT trong cùng 1 kết quả trả về

VD:

SELECT a, b FROM table1 UNION SELECT c, d FROM table2

Truy vấn SQL này sẽ trả về một tập hợp kết quả duy nhất có hai cột, chứa các giá trị từ cột avà btrong table1và cột cvà dtrong table2.

Để một UNIONtruy vấn hoạt động, hai yêu cầu chính phải được đáp ứng:

* Các truy vấn riêng lẻ phải trả về cùng một số cột.
* Các kiểu dữ liệu trong mỗi cột phải tương thích giữa các truy vấn riêng lẻ.

Để thực hiện một cuộc tấn công SQL injection UNION, bạn cần đảm bảo rằng cuộc tấn công của bạn đáp ứng hai yêu cầu này. Điều này thường liên quan đến việc tìm ra:

* Có bao nhiêu cột đang được trả về từ truy vấn ban đầu?
* Những cột nào được trả về từ truy vấn ban đầu thuộc loại dữ liệu phù hợp để giữ kết quả từ truy vấn được đưa vào?

Các Hackers có thể dùng các kết quả đó để lấy cắp các thông tin,….

* + - **Out-of-band SQLi**

Xảy ra khi kẻ tấn công không thể trực tiếp tấn công và thu thập kết quả trực tiếp trên cùng một kênh (server trả lại những response không mang thông tin có thể giúp kẻ tấn công tiếp tục thực hiện được quá trình tấn công)

Kiểu tấn công này bắt server thực hiện các request DNS hoặc HTTP để chuyển dữ liệu cho kẻ tấn công

Vd: Hãy xem xét URL sau do kẻ tấn công tạo ra.



Truy vấn thứ hai đang gọi một thủ tục được lưu trữ xp\_dirtree. Thủ tục được lưu trữ mở rộng này có thể được sử dụng để lấy danh sách tất cả các thư mục cho thư mục có tên trong xp.

SQL Server bây giờ sẽ tiến hành liệt kê tất cả các thư mục từ \\ test.attacker.com \ . Để thực hiện việc này, trước tiên nó phải giải quyết địa chỉ của tên miền test.attacker.com , nơi nó thực hiện truy vấn DNS tới máy chủ DNS của kẻ tấn công. Kẻ tấn công có thể theo dõi nhật ký máy chủ DNS và tìm kiếm các truy vấn tới test.attacker.com . Nếu một truy vấn DNS như vậy được thực hiện, điều đó có nghĩa là lỗ hổng SQL Injection có thể bị khai thác thông qua vector Out-of-Band.

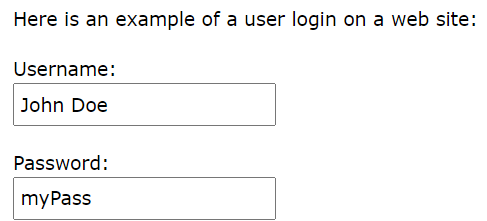
* + - **Blind SQLi**

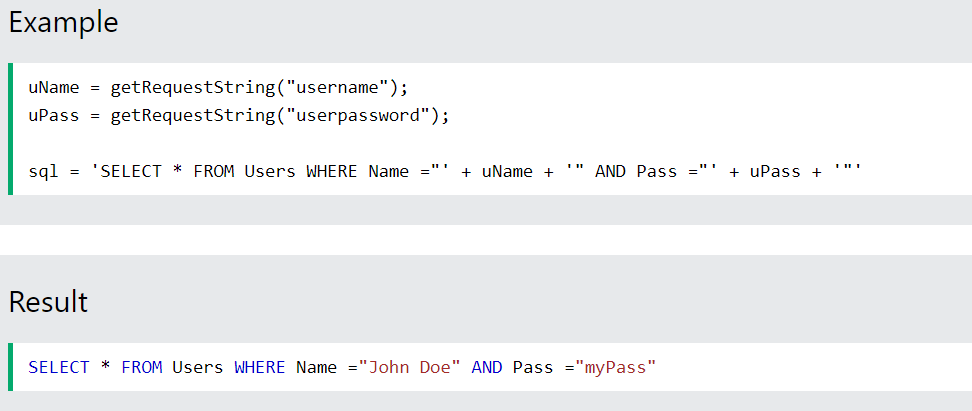
  Là kiểu tấn công sử dụng những đoạn mã độc SQL mà kết quả của đoạn mã không được hiển thị trực tiếp ở ứng dụng, tuy nhiên kẻ tấn công có thể suy luận được kết quả của đoạn mã

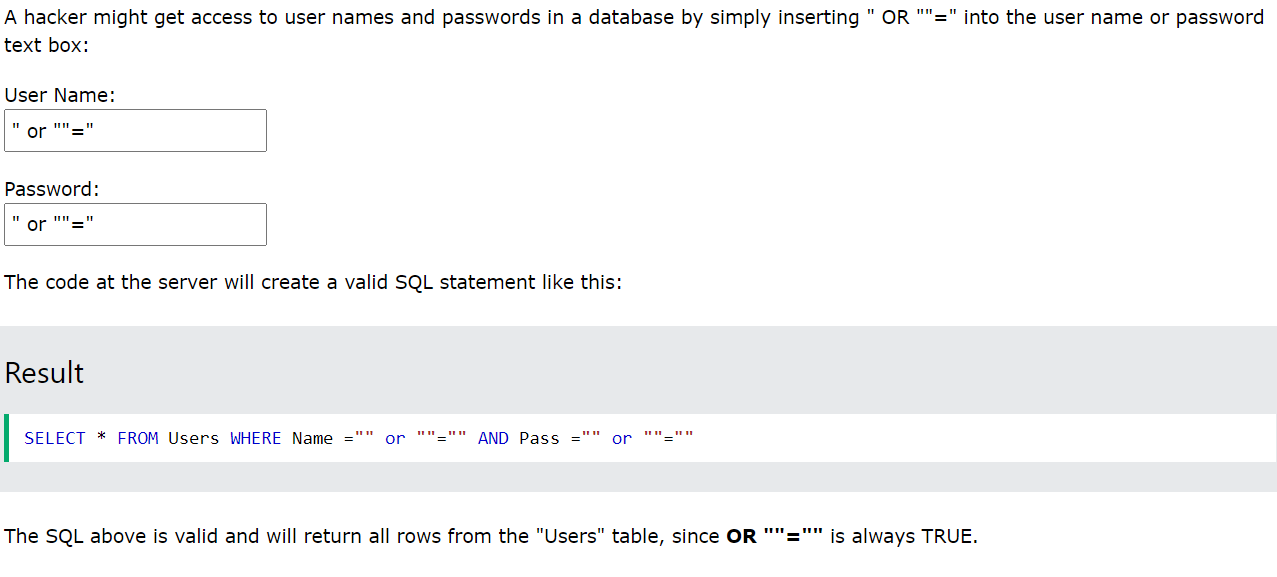
* + - * 1. **Boolean-based**

Là kĩ thuật tấn công SQL Injection dựa vào việc gửi những câu truy vấn tới DB và buộc ứng dụng trả về các kết quả khác nhau phụ thuộc vào câu truy vấn là True hay False

VD:SQL Injection Based on ""="" is Always True







* + - * 1. **Time-based**

Là kĩ thuật tấn công dựa vào việc gửi những câu truy vấn tới DB và buộc DB phải chờ một khoảng thời gian trước khi phản hồi

Thời gian phản hồi sẽ cho phép kẻ tấn công suy đoán kết quả truy vấn là True hay False

VD:



* 1. **Phòng ngừa hổng SQL Injection**
     + Tham số hóa truy vấn

Sử dụng PreparedStatement

Text

Description automatically generated

Sử dụng Stored Procedure

Text

Description automatically generated

* + - Xác thực input của người dùng

Lọc bỏ các ký tự như ‘, “, --

* + - Không hiển thị thông báo lỗi SQL cho người dùng
    - Mã hóa dữ liệu được lưu trong DB
    - Thiết lập đặc quyền phù hợp cho DB user

Không nên sử dụng user root của DBMS cho ứng dụng

Nếu 1 ứng dụng chỉ cần sử dụng các câu SELECT thì không cần thiết phải cấp quyền thực thi các câu lệnh INSERT, DELETE, UPDATE cho user

* Thực hiện pentest để tìm ra các lỗ hổng