

HỌC VIỆN KỸ THUẬT MẬT MÃ

KHOA AN TOÀN THÔNG TIN



ĐỒ ÁN MÔN HỌC
MÔN XÂY DỰNG ỨNG DỤNG WEB

ĐỀ TÀI: LẬP TRÌNH CÁC KỊCH BẢN TẤN CÔNG
SQL INJECTION VÀ CÁC PHƯƠNG PHÁP
PHÒNG CHỐNG

Giảng viên hướng dẫn: Hoàng Thanh Nam

Sinh viên thực hiện: Phạm Trần Sang

Trần Hữu Anh Văn

Huỳnh Phương Nam

Diệp Hân

Nguyễn Thị Xuân Lan

MỤC LỤC

CHƯƠNG 1: KỸ THUẬT TẤN CÔNG SQL INJECTION VÀ CÁC CÁCH TẤN CÔNG PHỔ BIẾN.....	1
1. SQL Injection là gì ?	1
<i>a. Định nghĩa.....</i>	<i>1</i>
<i>b. Hậu quả của SQL Injection.....</i>	<i>2</i>
2. Các dạng tấn công SQL Injection	2
<i>a. Dạng tấn công vượt qua kiểm tra lúc đăng nhập</i>	<i>2</i>
<i>b. Tấn công sử dụng câu lệnh SELECT</i>	<i>4</i>
<i>c. Tấn công sử dụng câu lệnh INSERT</i>	<i>4</i>
<i>d. Tấn công sử dụng stored-procedures</i>	<i>5</i>
CHƯƠNG 2: PHÒNG CHỐNG SQL INJECTION	6
1. Đối với website (dành cho lập trình viên)	6
2. Đối với web server (dành cho quản trị mạng)	7
3. Đối với database server (dành cho quản trị mạng)	7
4. Hạn chế bị phát hiện lỗi	7
5. Phòng chống từ bên ngoài.....	8
6. Cải thiện dữ liệu nhập vào	9
7. Một số lời khuyên khác.....	10
8. Một số công cụ quét và kiểm tra lỗi SQL Injection hiệu quả	10
CHƯƠNG 3: DEMO TRIỂN KHAI TẤN CÔNG SQL INJECTION.....	12
1. Chuẩn bị	12
2. Triển khai tấn công	12
3. Phòng tránh	14
TÀI LIỆU THAM KHẢO	17

LỜI NÓI ĐẦU

Với bùng nổ internet kèm theo phát triển Word Wide Web. Các doanh nghiệp, cá nhân phủ phát trang web hay ứng dụng web cung cấp đầy đủ giải pháp hiệu quả, đáng tin cậy, thách thức giao tiếp tiến hành thương mại hóa kỷ XXI. Tuy nhiên, an toàn trang web hay ứng dụng web trở nên ngày càng quan trọng thập kỷ qua. Ngày nay, trang web giáo dục, y tế, tài liệu nhạy cảm phải đối mặt với nhiều nguy cơ bị tấn công từ hacker. Tại Việt Nam năm vừa qua có nhiều tấn công nhắm vào tổ chức lớn gây thiệt hại nhiều cho doanh nghiệp, tổ chức. Nổi bật tấn công vào trang chủ VietnamAirlines nhóm hacker có tên 1937CN từ Trung Quốc gây chú ý nhiều dư luận.

Nhiều lỗ hổng trang web không kiểm tra kỹ để điều khiển ứng dụng trang web nguyên nhân để hacker dựa vào để tấn công SQL Injection dạng tấn công phổ biến sử dụng.

Ngoài ra có số dạng tấn công khác như: Shell Injection, Script language injection, file inclusion, XML injection, XPATH injection SQL Injection dạng công nghệ tấn công vào trang web. Với việc lợi dụng lỗ hổng câu lệnh truy vấn, hacker thêm vào số câu lệnh truy vấn SQL để chiếm quyền truy cập để thay đổi thông tin.

Nội dung báo cáo được trình bày sẽ xoay quanh ba nội dung chính:

- Chương 1: SQL Injection & các cách tấn công phổ biến.
- Chương 2: Phòng chống SQL Injection.
- Chương 3: Kịch bản tấn công & phương pháp phòng chống.

CHƯƠNG 1: KỸ THUẬT TẤN CÔNG SQL INJECTION VÀ CÁC CÁCH TẤN CÔNG PHỔ BIẾN



Hình 1.1: Mô hình tấn công SQL Injection

1. SQL Injection là gì ?

a. Định nghĩa

- SQL Injection là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL bất hợp pháp.

- SQL Injection có thể cho phép những kẻ tấn công thực hiện các thao tác như một người quản trị web trên cơ sở dữ liệu của ứng dụng như select, insert, update, ..., thậm chí là server mà ứng dụng đó đang chạy. Thông qua đó, hacker có thể lấy được các thông tin quan trọng từ website như username, password của trang quản trị, các tài khoản của khách hàng, ...

- SQL Injection thường được biết đến như là một vật trung gian tấn công trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu có cấu trúc như SQL Server, MySQL, Oracle, DB2, Sysbase, ...

- Công cụ dùng để tấn công là một trình duyệt web bất kỳ chẳng hạn như Internet Explorer, Netscape, Lynx, Firefox, ...

b. Hậu quả của SQL Injection

- Làm lộ dữ liệu trong database. Tùy vào tầm quan trọng của dữ liệu mà hậu quả dao động ở mức nhẹ cho đến vô cùng nghiêm trọng. Đây là hậu quả lớn nhất mà SQL Injection gây ra.

- Lộ dữ liệu khách hàng có thể ảnh hưởng rất nghiêm trọng đến công ty. Hình ảnh công ty cũng sẽ bị ảnh hưởng, dễ dẫn đến phá sản, ...

- Lỗ hổng này cũng ảnh hưởng đến khách hàng vì họ thường dùng chung một mật khẩu cho nhiều tài khoản, chỉ cần lộ mật khẩu một tài khoản thì các tài khoản khác cũng lộ theo.

- Trong nhiều trường hợp, hacker không chỉ đọc được dữ liệu mà còn có thể chỉnh sửa dữ liệu. Khi đó, hacker có thể đăng nhập dưới vai trò admin, lợi dụng hệ thống, hoặc xóa toàn bộ dữ liệu để hệ thống ngừng hoạt động.

2. Các dạng tấn công SQL Injection

Có bốn dạng thông thường bao gồm:

- Vượt qua kiểm tra lúc đăng nhập (Aauthorization bypass)
- Sử dụng câu lệnh SELECT
- Sử dụng câu lệnh INSERT
- Sử dụng các stored-procedures

a. Dạng tấn công vượt qua kiểm tra lúc đăng nhập

- Cách thức tấn công: Hacker 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.

- Thông thường để cho phép người dùng truy cập vào các trang web được bảo mật, hệ thống thường xây dựng trang đăng nhập để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu.

- Sau khi người dùng nhập thông tin vào, hệ thống sẽ kiểm tra tên đăng nhập và mật khẩu có hợp lệ hay không để quyết định cho phép hay từ chối thực hiện tiếp.

```
login.htm
<form action="ExecLogin.asp" method="post">
  Username: <input type="text" name="fUSRNAME"><br>
  Password: <input type="password" name="fPASSWORD"><br>
  <input type="submit">
</form>
```

Hình 1.2

```
execlogin.asp
<%
  Dim vUserName, vPassword, objRS, strSQL
  vUserName = Request.Form("fUSRNAME")
  vPassword = Request.Form("fPASSWORD")

  strSQL = "SELECT * FROM T_USERS " & _
           "WHERE USR_NAME=' " & vUserName & _
           "' and USR_PASSWORD=' " & vPassword & "' "

  Set objRS = Server.CreateObject("ADODB.Recordset")
  objRS.Open strSQL, "DSN=..."

  If (objRS.EOF) Then
    Response.Write "Invalid login."
  Else
    Response.Write "You are logged in as " & objRS("USR_NAME")
  End If

  Set objRS = Nothing
%>
```

Hình 1.3 Tấn công vượt qua kiểm tra đăng nhập

Ví dụ: Nếu hacker nhập vào chuỗi ‘ OR 1=1 Câu truy vấn sẽ là:

SELECT * FROM T_USERS WHERE USR_NAME =” OR 1=1 and
USR_PASSWORD=” OR 1=1

1. Kiểm tra USR_NAME=” đúng hay sai? → Sai
2. Kiểm tra 1=1 đúng hay sai? → Đúng
3. Kiểm tra USR_PASWROD=” đúng hay sai? → Sai
4. Kiểm tra 1=1 đúng hay sai? → Đúng

(1) or (2) -> Đúng, (3) or (4) -> Đúng

- Câu truy vấn này là hợp lệ và sẽ trả về tất cả các bản ghi của T_USERS và đoạn mã tiếp theo xử lý người dùng đăng nhập bất hợp pháp này như là người dùng đăng nhập hợp lệ.

- Tương tự như vậy, hacker có thể lợi dụng và làm những điều tồi tệ hơn trên

chính website của bạn,

b. Tấn công sử dụng câu lệnh SELECT

- Cách thức tấn công: Để thực hiện được kiểu tấn công này, hacker 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 này phức tạp hơn.

- Thông thường, sẽ có một trang nhận ID của tin cần hiển thị rồi sau đó truy vấn nội dung của tin có ID này.

- Khi đó hacker sẽ lợi dụng lỗi của site sẽ khai thác và lấy thông tin như: table, columns..., hoặc hiệu chỉnh, xóa dữ liệu bằng các câu lệnh SQL.

- Tấn công kiểu select này tuy phức tạp nhưng thường được hacker sử dụng, hacker thường khai thác lỗi này để lấy cấp tài khoản chùa hoặc chiếm quyền Admin của một webstie nào đó.

- Trong một trang web thương mại điện tử khách hàng chọn một sản phẩm và thông tin của sản phẩm đó được hiển thị qua trang web item.php qua câu truy vấn sau:

```
SELECT * FROM items WHERE 'id'=1;
```

Thông thường, đoạn mã này hiển thị nội dung của tin có ID trùng với ID đã chỉ định và hầu như không thấy có lỗi.

Tuy nhiên, nếu thay thế một ID hợp lệ bằng cách gán ID cho một giá trị khác ví dụ như: and 1=1

Câu truy vấn SQL lúc này sẽ trả về item có id=10 và câu lệnh này hoàn toàn chạy một cách chính xác theo yêu cầu hiển thị:

```
SELECT * FROM 'items' WHERE 'id'=10 and 1=1;
```

Tuy nhiên nếu thay thế id=1 or 1=1 thì trang web item.php sẽ hiển thị toàn bộ danh sách các item trong bảng items, rõ ràng đây là điều bất hợp lý vì trang item.php có tác dụng hiển thị thông tin của một món hàng mà khách hàng đã chọn.

c. Tấn công sử dụng câu lệnh INSERT

- Cách thức tấn công: 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 có thể được dùng khi hệ thống không kiểm tra tính hợp lệ của thông tin nhập vào.

- Bằng cách chèn các câu lệnh insert hacker có thể sử dụng thêm vào database các thông tin không mong muốn nếu trang web thu thập thông tin không kiểm tra tính chính xác khi xác minh.

- Một câu lệnh INSERT có thể có cú pháp dạng:

INSERT INTO TableName

VALUES ('Value One', 'Value Two', 'Value Three').

d. 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.

- Việc tấn công bằng stored-procedures sẽ gây tác hại rất lớn nếu ứng dụng được thực thi với quyền quản trị hệ thống 'sa'.

- Ví dụ 1, stored procedure splogin gồm hai tham số là username và password, nếu kẻ tấn công nhập: Username:thanhcong Password; Shutdown—Lệnh gọi stored procedure như sau: EXEC splogin 'thanhcong',''; shutdown- - Lệnh shutdown thực hiện dừng sql server ngay lập tức.

- Ví dụ 2, nếu ta thay đoạn mã tiêm vào dạng: ' ; EXEC xp_cmdshell 'cmd.exe dir C: '. Lúc này hệ thống sẽ thực hiện lệnh liệt kê thư mục trên ổ đĩa C:\ cài đặt server. Việc phá hoại kiểu nào tùy thuộc vào câu lệnh đăng sau cmd.exe.

CHƯƠNG 2: PHÒNG CHỐNG SQL INJECTION

Lỗi SQL Injection là khá nguy hiểm vì thế việc phòng tránh là rất cần thiết. Loại bỏ các ký tự và từ khóa nguy hiểm như: --, select, where, drop, shutdown ...

Để phòng tránh các nguy cơ có thể xảy ra, hãy bảo vệ các câu lệnh SQL là bằng cách kiểm soát chặt chẽ tất cả các dữ liệu nhập nhận được từ đối tượng Request (Request, Request.QueryString),

Request.Form, Request.Cookies, and Request.ServerVariables).

Ví dụ, có thể giới hạn chiều dài của chuỗi nhập liệu, hoặc xây dựng hàm EscapeQuotes để thay thế các dấu nháy đơn bằng 2 dấu nháy đơn.

Cần có cơ chế kiểm soát chặt chẽ và giới hạn quyền xử lý dữ liệu đến tài khoản người dùng mà ứng dụng web đang sử dụng. Các ứng dụng thông thường nên tránh dùng đến các quyền như dbo hay sa. Quyền càng bị hạn chế, thiệt hại càng ít.

Ngoài ra để tránh các nguy cơ từ SQL Injection attack, nên chú ý loại bỏ bất kỳ thông tin kỹ thuật nào chứa trong thông điệp chuyển xuống cho người dùng khi ứng dụng có lỗi. Các thông báo lỗi thông thường tiết lộ các chi tiết kỹ thuật có thể cho phép kẻ tấn công biết được điểm yếu của hệ thống.

1. Đối với website (dành cho lập trình viên)

Cần kiểm tra tính đúng đắn của tất cả dữ liệu đầu vào. Dữ liệu đầu vào không chỉ là các tham số, mà bao gồm cả cookie, user agent, referer ...

Việc kiểm tra tính đúng đắn của dữ liệu có thể dựa trên các phương pháp sau:

- Kiểm tra dựa vào kiểu dữ liệu (số, ngày tháng ...)
- Kiểm tra giới hạn độ dài đầu vào:
 - + Loại bỏ các ký tự đặc biệt như: ' % " ? # @ & ...
 - + Loại bỏ các từ đặc biệt: select, drop, delete, information_schemal, insert, union, xp_ ...

2. Đối với web server (dành cho quản trị mạng)

Hầu hết các máy chủ web (web server) hiện nay đều có các module hỗ trợ việc phòng chống SQL Injection, ví dụ:

Apache có modsecurity, IIS có URLScan. Bạn chỉ cần bật tính năng này và cấu hình cho phù hợp. Nếu website của bạn là dạng trang tin tức thì rất phù hợp để triển khai. Trong một số trường hợp khác, các module này có thể chặn nhầm, dẫn tới website hoạt động không chính xác.

3. Đối với database server (dành cho quản trị mạng)

Bạn cần thực hiện việc cấu hình phân quyền chặt chẽ đối với các tài khoản. Khi đó, dù tồn tại lỗi SQL Injection, thiệt hại cũng sẽ được hạn chế. Ngoài ra, bạn cần loại bỏ các bảng, thành phần và tài khoản không cần thiết trong hệ thống.

4. Hạn chế bị phát hiện lỗi

Attacker dựa vào những lỗi trong lập trình ứng dụng để tấn công và cụ thể attacker dựa vào các dấu hiệu để phát hiện ứng dụng bị lỗi. Vậy việc làm cho các dấu hiệu đó bị che đi, trở nên khó hiểu hơn, hoặc biến mất... được hầu hết các chuyên gia bảo mật sử dụng. Lưu ý là kỹ thuật này chỉ dùng để dấu lỗi, còn lỗi trên ứng dụng vẫn còn đó, chỉ là để chống lại sự phát hiện quá dễ dàng lỗi để kẻ xấu khai thác.

Nhưng những attacker khôn khéo vẫn có thể nhìn thấu được kiểu phòng chống như thế này. Nó có thể tránh được những tấn công đơn giản như là thêm dấu '(dấu nháy)' vào cuối đường dẫn. Vì phương pháp tìm kiếm ứng dụng bị lỗi của những tấn công như thế dựa vào những dấu hiệu trả về của ứng dụng hoặc trực tiếp từ database. Ta có thể chỉ đưa ra những thông báo chung chung hoặc định hướng trở lại trang ban đầu(redirect). Trong trường hợp này, công việc tìm kiếm lỗi và xác định mục tiêu trở nên cực khó đối với attacker.

Tuy nhiên attacker luôn tạo ra những công nghệ tìm kiếm lỗi tinh vi hơn, tốt hơn, để gián tiếp xác định dấu hiệu trả về. Tấn công kiểu này còn được gọi là “Blind SQL Injection”

5. Phòng chống từ bên ngoài

Giải pháp này sẽ dùng tường lửa đặc biệt để bảo vệ bạn khỏi những ứng dụng dùng việc truy cập database với mục đích xấu. Chúng ta cần lưu ý rằng attacker tương tác với ứng dụng web thông qua một trình duyệt với kết nối từ xa. Sau đó, ứng dụng gửi yêu cầu đến database. Như vậy chúng ta có thể ngăn chặn các tấn công giữa attacker với ứng dụng, giữa ứng dụng với database và ngay cả trên chính bản thân database đó.

Một số phương pháp phòng chống có thể thực hiện như:

Cách phòng chống	Vị trí	Mô tả	Điểm yếu
Web shields	Kiểm soát giữa người dùng và ứng dụng web	Lọc ra những yêu cầu với đường dẫn khả nghi gửi đến ứng dụng web nhằm ngăn chặn tấn công trước khi nó được đưa đến ứng dụng.	Attacker vẫn có thể thử bằng nhiều cách khác nhau để tìm ra một cách có thể đánh lừa được cách phòng chống này. Các dạng tấn công càng ngày càng tinh vi và khôn khéo nên nó sẽ bị đánh bại nếu gặp một tấn công mà nó chưa được “học”.
Web scanners	Kiểm soát giữa người dùng và ứng dụng web	Tự trực tiếp tấn công hoặc dùng những công cụ giả tấn công, kiểm tra trạng thái hoạt động để lấy cơ sở cấu hình lại	Như trên
SQL shields	Kiểm soát giữa ứng dụng và database	Tương tự như web shield, cách này sẽ kiểm tra tất cả các lưu lượng truy vấn và phân tích nó sử dụng những dạng tín hiệu hoặc những dạng bất bình thường để xác định những truy vấn có hại.	Như trên
Database access controls	Kiểm soát trên database	Chỉ cho phép đặc quyền cần thiết tối thiểu cho ứng dụng để không có bất cứ truy cập vượt quyền nào có thể truy cập được.	Thông thường việc điều khiển truy cập vào cơ sở dữ liệu không đủ để ngăn chặn tất cả các cuộc tấn công.

Những bộ lọc, bộ quét và những điều khiển truy cập cơ sở dữ liệu sẽ làm cho ứng dụng web khó bị tấn công hơn.

6. *Cải thiện dữ liệu nhập vào*

Cách phòng chống thực sự để chống lại SQL Injection là kiểm tra và làm đúng các câu truy vấn. Như chúng ta đã đề cập, lỗi này là do ứng dụng không kiểm tra dữ liệu nhập vào của người dùng. Do đó người dùng có thể thay đổi, chỉnh sửa, tham số hoặc thêm cả một thực thể truy vấn vào câu lệnh. Vì thế mỗi dữ liệu nhập của người dùng cần được theo dõi và có những ràng buộc nhất định.

Thứ nhất, ứng dụng cần phân loại các kiểu dữ liệu nhập vào. Ví dụ, nếu ứng dụng yêu cầu dữ liệu nhập vào là kiểu số thì khi ứng dụng nhận dữ liệu nhập vào không nên chấp nhận các kiểu khác ngoại trừ kiểu số. Một số hàm kiểm tra trong PHP:

`is_numeric($str)`: kiểm tra \$str có phải kiểu số hay không
`is_int($str)`: kiểm tra kiểu interger
`is_float($str)`: kiểm tra kiểu số thực
...

Thứ 2, nếu dữ liệu nhập vào không rõ kiểu gì thì ít nhất cũng phải xác định những kiểu không được phép có thể được gọi. Trong trường hợp này chúng ta sẽ phải lọc các dấu nháy, lệnh, các kí tự đặc biệt. Một vài việc lọc dữ liệu có thể thực hiện trên toàn bộ ứng dụng (như không bao giờ lưu dữ liệu có dấu ‘ vào cơ sở dữ liệu) và trên một vài kiểu dữ liệu nhập vào(như không có dấu “,” trong địa chỉ mail).

Ví dụ:

`magic_quotes_gpc`

⁹
`GPC=GET,POST,COOKIE)` Hàm sẽ kiểm tra các dữ liệu thuộc 3 loại trên và khi phát hiện có các dấu ‘ (single-quote), ” (double quote),

Trong khi viết một cơ sở dữ liệu hướng ứng dụng, hay khi triển khai một ứng dụng mã nguồn mở cần chú ý đến các vấn đề như thế và thiết kế để xác minh đúng đầu vào. Biện pháp này sẽ giúp bảo vệ bạn từ các tấn công SQL Injection không trở thành môi ngon cho các attacker.

Hiểu biết về cách phòng chống này là rất quan trọng nếu bạn đang triển khai một ứng dụng thương mại. Chỉ cần nhớ rằng các nhà phát triển có khả năng vướng lỗi khi lập trình và bạn sẽ phải thực hiện các bước để sửa các lỗi đó. Và cần làm điều này ngay cả khi chưa có những lỗ hổng được công khai cho ứng dụng đó.

7. Một số lời khuyên khác

- Bạn cần tắt tất cả các thông báo lỗi không cần thiết của web server. Hacker có thể lợi dụng chính các thông báo lỗi này để khai thác thông tin của hệ thống, phục vụ cho một cuộc tấn công SQL Injection

- Bạn cần bật các chế độ ghi log đầy đủ để phục vụ việc điều tra phát hiện cuộc tấn công và giải quyết sự cố.

- Bạn cần thường xuyên theo dõi và cập nhật phiên bản cho platform của website (hệ điều hành, web server, database server ...).

8. Một số công cụ quét và kiểm tra lỗi SQL Injection hiệu quả

Acunetix Web Vulnerability Scanner: Một phiên bản thương mại của chương trình tìm kiếm các lỗ hổng bảo mật trên các ứng dụng Web. Acunetix WVS tự động kiểm

tra các ứng dụng Web để tìm kiếm các lỗ hổng bảo mật như SQL Injection, hay Cross-Site Scripting, tìm kiếm những chính sách đối với mật khẩu đăng nhập cũng như các phương thức xác thực vào Web Site. Với giao diện đồ họa thân thiện, những Report đầy đủ cho phép bạn kiểm tra những vấn đề trên máy chủ và ứng dụng Web. Để tìm hiểu rõ hơn các bạn có thể truy cập vào <http://www.acunetix.com/vulnerability-scanner/> để tải về.

N-Stealth: Là một phiên bản thương mại, ứng dụng cho việc tìm kiếm các lỗ hổng bảo mật trên máy chủ Web. Phần mềm tự động update thường xuyên hơn các phần mềm miễn phí như Whisker/libwhisker hay Nikto, nhưng nhiều lỗi mới trên Web cũng không phát hiện kịp thời và nhanh chóng. Phần mềm bao gồm hơn 30.000 lỗ hổng có thể Scan và khai thác trực tiếp, cùng với hàng tá những cập nhật hàng ngày. Dễ dàng triển khai kết hợp với những Scan lỗ hổng bảo mật như: SQL Injection, Nessus, ISS Internet Scanner, Retina, SAINT và Sara, bao gồm các tính năng khác. N-sealth là phiên bản chỉ dành riêng cho Windows và không thể download Source.

CHƯƠNG 3: DEMO TRIỂN KHAI TẤN CÔNG SQL INJECTION

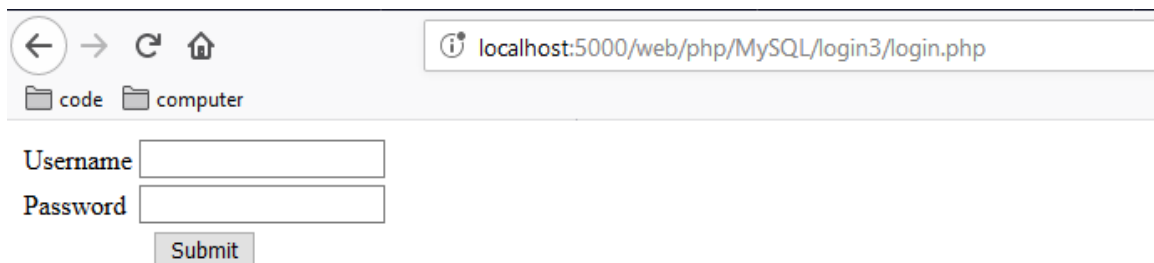
1. Chuẩn bị

- XAMPP, PHP, MySQL
- IDE: notepad, sublime text, visual code, ...
- Browser

2. Triển khai tấn công

```
$username = $_POST['username'];  
$password = $_POST['password'];  
$query = "SELECT username,password from users where username = '".  
    username.'" and password = '". $password."";  
$result = mysqli_query($conn, $query) or die (mysqli_error());  
$count = mysqli_num_rows($result);  
  
if($count == 0){  
    $msg = 'Username and password is not correct, please try again!';  
    form();  
} else {  
    $msg = 'Login Success';  
    form();  
}
```

Hình 3.1: Câu lệnh thực thi



← → ↻ 🏠 localhost:5000/web/php/MySQL/login3/login.php

📁 code 📁 computer

Username

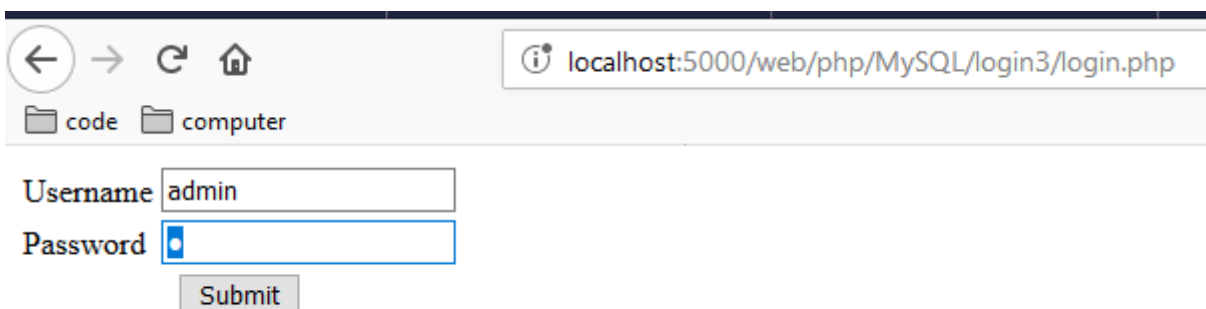
Password

Submit

Hình 3.2: Giao diện đăng nhập

- Tấn công:

+ Check lỗi sql injection, sử dụng tài khoản với tên đăng nhập là: admin.
Phần password sử dụng kí tự nhảy đơn để check lỗi sql injection



← → ↻ 🏠 localhost:5000/web/php/MySQL/login3/login.php

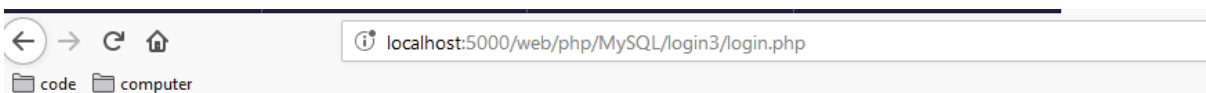
📁 code 📁 computer

Username

Password

Submit

Hình 3.3: Kiểm tra đầu vào



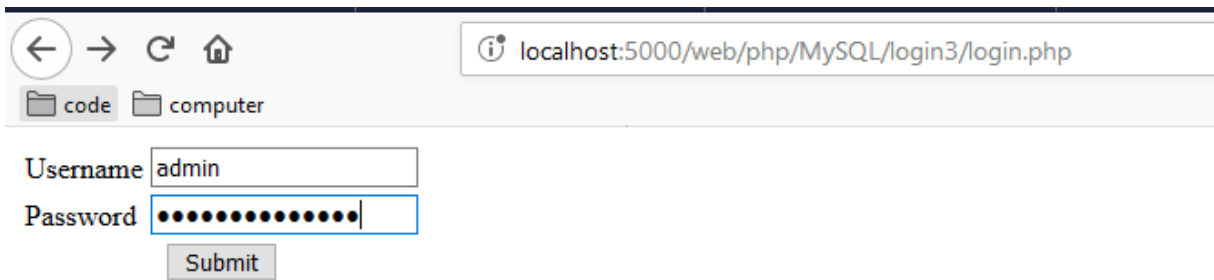
← → ↻ 🏠 localhost:5000/web/php/MySQL/login3/login.php

📁 code 📁 computer

Warning: mysqli_error() expects exactly 1 parameter, 0 given in C:\xampp-1\htdocs\web\php\MySQL\login3\login.php on line 16

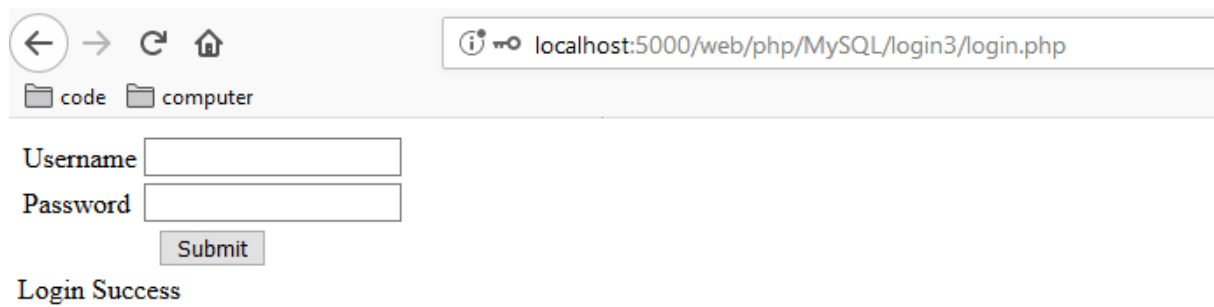
Hình 3.4: Thông báo lỗi đầu vào

- + Có thông báo lỗi => có thể có lỗi sql injection
- Sử dụng lệnh: admin' or '1' = '1 để tấn công



Hình 3.5: Tiến hành khai thác lỗi sql injection

- Sau khi bấm submit thì nó báo login thành công



Hình 3.6: Login thành công

3. Phòng tránh

- Có 4 cách phòng tránh cơ bản

+ Cách 1: Sử dụng Prepared Statement

```
$username = $_POST['username'];
$password = $_POST['password'];
$stmt = $conn->prepare("SELECT username,password FROM users WHERE username =? AND password = ?");
$stmt->bind_param('ss',$username,$password);
$stmt->execute();
$stmt->bind_result($username,$password);
$stmt->store_result();
if($stmt->num_rows==0){
    $msg='Username and password is not correct, please try again!';
    form();
} else {
    $msg = 'Login Success';
    form();
}
$stmt->close();
$conn->close();
```

+ Cách 2: Sử dụng mysqli_real_escape_string

```
$username = mysqli_real_escape_string($conn,$_POST['username']);
$password = mysqli_real_escape_string($conn,$_POST['password']);
$query = "SELECT username,password from users where username = '". $username."' and password = '".
    $password."'";
$result = mysqli_query($conn, $query) or die (mysqli_error());
$count = mysqli_num_rows($result);
if($count == 0){
    $msg = 'Username and password is not correct, please try again!';
    form();
} else {
    $msg = 'Login Success';
    form();
}
mysqli_close($conn);
```

+ Cách 3: Sử dụng strip_tags

```
$username = strip_tags(trim($_POST['username']));
$password = strip_tags(trim($_POST['password']));
$query = "SELECT username,password from users where username = '". $username."' and password = '".
    $password."'";
$result = mysqli_query($conn, $query) or die (mysqli_error());
$count = mysqli_num_rows($result);
if($count == 0){
    $msg = 'Username and password is not correct, please try again!';
    form();
} else {
    $msg = 'Login Success';
    form();
}
mysqli_close($conn);
```

+ Cách 4: Sử dụng PDO

TÀI LIỆU THAM KHẢO

<https://www.lionblogger.com/ways-to-prevent-sql-injection-attacks-in-php/>

https://www.w3schools.com/php/php_ref_mysql.asp