Fireproof

One Password to Rule Them All

Annie Kelly, Tyler Tafoya, Kara James, Tori Augustine
https://github.com/shadowfax6894/CS3308

Automated Test Cases

For our automated testing we utilized the Python unittest framework. We split up our testing into two categories: testing account creation/credentials, and testing the insertion/retrieval of information from our MySQL database.

Testing the creation of MasterAccount objects

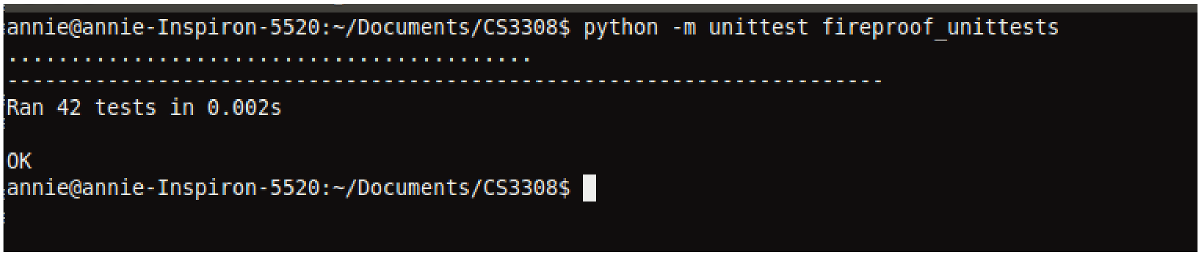
Our tests included:

- ensuring that the MasterAccount() constructor intialized correctly by:
  - asserting that the object was an instance of MasterAccount
  - asserting that the username given to the constructor was initalized correctly and accessible using the .username attribute
  - asserting that the password given to the constructor was initialized correctly and accessible using the .password attribute
- ensuring that our encryption suite worked properly by:
  - asserting that the username given to the constructor was not equal to the username after encryption
  - asserting that the password given to the constructor was not equal to the password after encryption
  - asserting that when we decrypt the encrypted username using the same symmetric key and initialization vector, that it is equal to the username that was originally given to the constructor
  - asserting that when we decrypt the encrypted password using the same symmetric key and initialization vector, that it is equal to the password that was originally given to the constructor

Instructions to run:

- Clone entire git repository
- Must have mysql-server, python-mysqldb, and python-tk packages installed
- run “python -m unittest fireproof_unittests”

Output:



```
annie@annie-Inspiron-5520:~/Documents/CS3308$ python -m unittest fireproof_unittests
..........................................
----------------------------------------------------------------------
Ran 42 tests in 0.002s

OK
annie@annie-Inspiron-5520:~/Documents/CS3308$ 
```

Testing the insertion and selection of accounts in our database

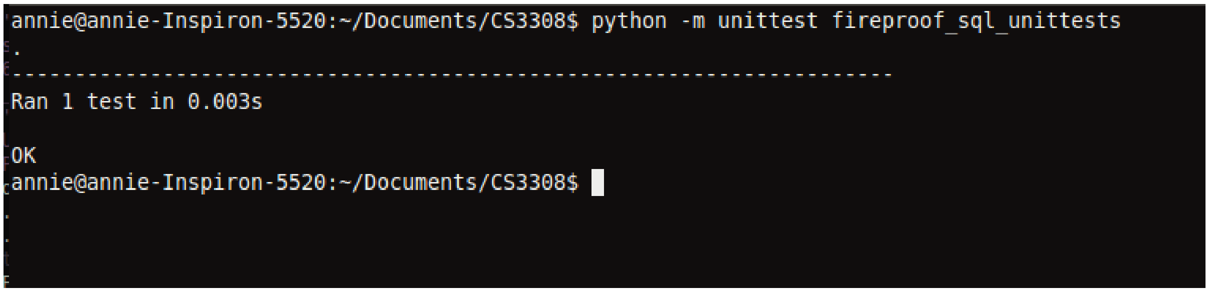
Our tests included:

- inserting 6 different MasterAccounts into our MySQL database and then asserting that when we retrieve an account from the database that it does not return None

Instructions to run:

- Clone entire git repository
- Must have mysql-server, python-mysqldb, and python-tk packages installed
- Running this testfile requires some extra steps:
    - launch MySQL by running "mysql -u root -p"
    - log in with root password
    - In the MySQL prompt run “CREATE DATABASE Fireproof;”
    - in the MySQL prompt run “CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'testAnnie';”
    - next run “GRANT ALL PRIVILEGES ON `%`.* TO 'testuser'@'localhost' IDENTIFIED BY 'testAnnie' WITH GRANT OPTION;”
- run “python -m unittest fireproof_sql_unittests”

Output:



```
annie@annie-Inspiron-5520:~/Documents/CS3308$ python -m unittest fireproof_sql_unittests
.
----------------------------------------------------------------------
Ran 1 test in 0.003s

OK
annie@annie-Inspiron-5520:~/Documents/CS3308$ 
```

User Acceptance Testing

Test Case ID: 1
Test Priority (Low/Medium/High): High
Module Name: Create a new account
Description: Create a new account by providing username and password on the sign up page

Preconditions: Username and passwords fields must be non-Null, password must be 8+ characters, password and confirm password fields must match
Dependencies: MySQL database, MasterAccount.py methods, Python Tkinter

Step	**Test Data**	**Result**	**Status**
Click “Sign Up” on start screen		Create Account screen appears	Pass

Fill out username, password, and confirm password fields	username = BillyBob
	password = Password
	confirmpassword = Password
		Username can be seen in the field as plaintext, while the password is seen as *’s	Pass

Click “Create Account” button		Username and password get encrypted using the master password and get inserted in the database	Pass

Test Case ID: 2
Test Priority (Low/Medium/High): High
Module Name: Logging into account
Description: Login into account by verifying username and password on the login page

Preconditions: Must have previously created an account
Dependencies: MySQL database, MasterAccount.py methods, Python Tkinter

Step	Test Data	Result	Status

Fill out username and password fields	username = BillyBob
	password = Password
		Username can be seen in the field as plaintext, while the password is seen as *'s	Pass

Click "Login" button		User is navigated to a new page	Pass

Test Case ID: 3
Test Priority (Low/Medium/High): Medium
Module Name: Go Back button
Description: Clicking "Go Back" on the Create Account page in order to get back to the Login page

Preconditions:
Dependencies: Python Tkinter

Step	Test Data	Result	Status
Click "Sign Up" on start screen		Create Account screen appears	Pass
Click "Go Back" button		User is navigated back to Login page	Pass

Test Case ID: 4

Test Priority (Low/Medium/High): Medium
Module Name: Account not found
Description: User tries to log into account that does not exist

Preconditions: Account must not exist
Dependencies: MySQL database, MasterAccount.py methods, Python Tkinter

Step	**Test Data**	**Result**	**Status**
Fill out username and password fields	Invalid account info	Username can be seen in the field as plaintext, while the password is seen as *'s	Pass
Click "Login" button		A window pops up that says "Username and password not found. Please click create an account" and a button that says "OK"	Pass
Click "OK" button		Popup window closes	Pass

Test Case ID: 5
Test Priority (Low/Medium/High): Medium
Module Name: Passwords don't match
Description: While a user is creating an account the "password" and "confirm password" fields do not match

Preconditions: Password fields must not match
Dependencies: MasterAccount.py methods, Python Tkinter

Step	**Test Data**	**Result**	**Status**
Click "Sign Up" on start screen		Create Account screen appears	Pass
Fill out username and password fields	Non-matching passwords	Username can be seen in the field as plaintext, while the password is seen as *'s	Pass

Click “Create Account” button		A window pops up that says “Passwords entered do not match.” and a button that says “OK”	Pass
Click “OK” button		Popup window closes	Pass

Test Case ID: 6
Test Priority (Low/Medium/High): Medium
Module Name: Passwords too short
Description: While a user is creating an account their password is less than 8 characters

Preconditions: Password must be < 8 characters
Dependencies: MasterAccount.py methods, Python Tkinter

Step	**Test Data**	**Result**	**Status**
Click “Sign Up” on start screen		Create Account screen appears	Pass
Fill out username and password fields	Password of length < 8	Username can be seen in the field as plaintext, while the password is seen as *’s	Pass
Click “Create Account” button		A window pops up that says “Password must be at least 8 characters.” and a button that says “OK”	Pass
Click “OK” button		Popup window closes	Pass