CSCI 362 Deliverable 5: SugarLabs

The Chocolate L'Eclercs

Alex Skiff, Blaine Billings, Carson Barber
Chase Myers, Justin Willis

1 Deliverable 5: Testing Fault-Injected Code

1.1 Documentation

When you first clone the directory, The full sugar labs source code will be in the testing-framework/sugar directory, the working sugar labs files that we have not changed will be in the testing-framework/scripts/fix/ directory, and the faulty files will be in the testing-framework/scripts/break/ directory.

To inject the faulty code, you will run the script breakSugar.py from within the scripts directory. After breaking the code, you will need to run the fixSugar.py script to get the original code back in the working sugar labs directory. Because of the sugar labs installation process, the code that is executed is actually found in the path: /usr/local/lib/python2.7/dist-packages/jarabe/.

When you run the fix or break scripts, we will be compiling the broken or fixed code, and replacing the files within this distribution folder. It is also possible that you will need to be root when you run the fix and break scripts.

1.2 Changed Code

The Sugar Labs software has been edited, as described in the preceding paragraphs, in the following source files with changes as described:

- **agepicker.py**: Changed the _SECONDS_PER_YEAR constant to $(365 * 24 * 60 * 60 * 60)$ instead of $(365 * 24 * 60 * 60)$ so our age calculation will no longer return the correct value, causing the age_calculator test case.
- **viewhelp.py**: Changed the social-help-server string to social-help in the get_social_help_server method, so our help window will no longer be able to find the correct help web page, making the Help Website test case fail.
- **httprange.py** Removed the line self._size = self._result from the size method. This will make the size of the web request always return 0, so the users would never be able to tell if the http requests are even working. This will also cause our HTTP Size test case to fail for the method size.
- **friends.py**: In the has_buddy method for a Friends object, we changed the check for buddy.get_key() in self._friends to not in friends. This makes the system think that people who are not friends are friends and vice versa. This will cause the has_buddy method fail within our Buddy Check test case.
- **schoolserver.py**: In the _generate_serial_number() we added a line to seed the random number generator so that this serial number creator is

no longer random. In our test case we have the seed match what the system usually used, so now our Serial Number test case will fail

1.3 Team Evaluation

For this deliverable we have changed the code in multiple places inside of Sugar Labs system in order to make our testing framework fail on a few test cases. The changes could be removing method implementation, making mathematical mistakes and other changes to cause assertions to no longer be valid. This will help us further test out the framework to make sure that even with broken code, it does not cause our framework to crash. This wasn't too difficult to plan out, the hardest part was replacing the actual sugar labs code. After installation Sugar Labs installs everything to the python distributable folder on your computer with the compiled files. To accompany these installations, we wrote scripts files to compile our changed code and replace the files in the sugar install directory. We also wrote a script file to replace the original code.