

CSCI 362 Deliverable 4: SugarLabs  
The Chocolate L'Eclairs  
Alex Skiff, Blaine Billings, Justin Willis, Carson Barber, Chase Myers

## **Deliverable 5: Code Injection**

### **1. Description:**

- 1.1. 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

### **2. Changed Code:**

- 2.1. *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.
- 2.2. *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.
- 2.3. *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`.
- 2.4. *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.
- 2.5. *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

### **3. How to Run:**

- 3.1. 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.

- 3.2. 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/'`

- 3.3. 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.