Chapter 2: Test Plan

The Testing Process

Our team plans to perform a series of tests on several methods found in the utils.c sub-system of the Tor project. We specifically plan on testing the following methods (as of now, we may add more later):

- round_to_the_next_multiple_of
- tor_strndup_

To do this, we will need to develop a small driver for the util.c methods so that our framework can function and also build a testing framework of at least twenty-five tests based off the test cases listed in this document. Once the framework is completed, we will run the tests and gather data on whether the methods passed or failed each.

As a final part of our testing process, we will inject faults into some of our tests to force a failure and analyze the results of the failures.

Tested Items

- util.c
  - method: round_to_the_next_multiple_of
  - tor_strndup

Test Cases

Case #	***Method Tested***	***Inputs***	***Oracle***
1	round_to_the_next_multiple_of	13, 4	16
2	round_to_the_next_multiple_of	0, 42	0
3	tor_strndup	abcdefg, 5	abcde
4	tor_strndup	qwerty, 0	(blank output)
5	tor_strndup	(empty), 5	(blank output)

Testing Schedule

Task	***Complete By***
Methods Driver Built	October 18th
Framework Built for First Five Test Cases	October 22nd
Full Framework Completed for All Twenty-Five Cases	November 12th
Report on Faults Injected into Test Cases	November 24th
Final Report on Tests	December 1st

Recording Procedures

We will keep a spreadsheet (which will be found on our wiki) to keep track of the results of each test. The spreadsheet rows will include the test case number, the method tested, the inputs tested, the expected output, the received output, and whether the test passed or failed. When tests fail, we will document the reason we believe to have failed (particularly in cases such as when we inject faults).

Constraints

Due to the amount of time we have to complete the testing framework, we do not expect to go over twenty-five test cases overall.

Further Comments

As of this deliverable, the team all feels as though we have taken on quite the undertaking in picking Tor as the project we're testing. Tor's primary difficulty comes from the sheer size of the project, but also the heavy amount of code related to encryption in its repository. We chose to test utils.c files because of their simplicity in comparison to the rest of the project and hope that we are correct that they won't be as difficult to test as the more complex methods for Tor. We are mostly optimistic that we'll be able to complete all 25 test cases if we stick to the helper methods.