# TEST PLAN

**20 April 2014**

**Team C**

**Jamie Lane, Bradley Norman, Daniel Ross**

# 1. Introduction

The test plan document is linked directly with the event-trace diagrams from the Team\_C\_ProjectDesign document dated 13 April 2014. Each event-trace diagram has a corresponding test plan listed in Table 1.

# 2. Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Number | Requirement Number | Test Description | Expected Result | Actual Result | Pass/Fail |
| 1 | 3 | Startup the application | The application’s GUI is opened and displayed. The GUI displays text fields where the user can enter a source IP address, destination IP address, and a port number. A transmit button is also available. The network connection is open and ready for use. |  |  |
| 2 | 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 | Enter a source IP address, destination IP address, and port number in GUI text fields. Click the transmit button. | The status bar displays a message stating, “Received Packet/Original Packet Ratio is: “ followed by the ratio of the packet sizes. |  |  |
| 3 | 3 | Network Unavailable on Start up. |  |  |  |
| 4 | 3, 5, 6, 7 | Invalid User Input |  |  |  |
| 5 | 3, 8, 9, 10, 11 | Packet Not Transmitted |  |  |  |
| 6 | 3, 8, 9, 10, 11, 12 | Packet Not Received |  |  |  |
| 7 | 3 | Shutdown the application by clicking the “X” at the top right hand corner of the GUI. | The application’s GUI is closed. The network connection is closed. |  |  |

**3. Requirements**

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| 1 | Shall be written using the Java v7 SDK and jNetPcap API |
| 2 | Shall run on the Oracle JVM (Java virtual machine), hosted on a currently supported version of the Microsoft Windows operating system |
| 3 | Shall provide the user with a graphical user interface |
| 4 | Shall run as a standalone application (neither as a client nor as a server) |
| 5 | Shall allow a source IP (internet protocol) address to be selected by the user as a target address |
| 6 | Shall allow a destination IP address (Open Arena server IP) to be selected by the user |
| 7 | Shall allow a destination port (Open Arena server port) to be selected by the user |
| 8 | Shall construct UDP (user datagram protocol) packets containing a message eliciting status from an Open Arena server |
| 9 | Shall construct IP packet headers, containing user selected addresses and ports |
| 10 | Shall combine IP packet headers and UDP packet payloads |
| 11 | Shall calculate complete packet-size, prior to transmission |
| 12 | Shall transmit packets to a selected Open Arena server, following user initiation |
| 13 | Shall receive packets from the selected Open Arena server |
| 14 | Shall calculate the size of received packets |
| 15 | Shall calculate the ratio of transmitted packet-size to received packet-size |
| 16 | Shall display the packet-size ratio (amplification ratio) |
| 17 | Shall not be operated remotely via any direct form of network control |
| 18 | Shall not be operated by an internal timer |
| 19 | Shall not obfuscate its operation via hidden user interface elements or deliberately opaque code |

|  |  |
| --- | --- |
|  | |
|  | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |