## **TauNet Software Test Plan v1.0**

Copyright © 2015 Gregory Gaston

Encrypt and Decrypt a string that contains all common keyboard characters. Test is uncuessful if the pre-ncryption and post-decryted strings are the same.  PAS decrypt a Ciphersaber 2 test string. Test is successful when decryption of 'Al Dakota buys' esults in 'mead'  Compose the message 'This is a protocol test' From: Sender, To Reciever, protocol v0.1 [Past is Successful if composed string is "From: Sender\tan\tan\tan\tan\tan\tan\tan\tan\tan\tan	Test:	RESULT:
Decrypt a Ciphersaber 2 test string. Test is successful when decryption of 'Al Dakota buys' esults in 'mead'  Compose the message 'This is a protocol test' From: Sender, To Reciever, protocol v0.1 [PAS Est is Successful if composed string is "From: Sender'n'nTo: Reciever'n'nVersion: 0.1\n'n\n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever'n'nVersion: 0.1\n'n\n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever'n'nVersion: 0.1\n'n\n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever'n'nVersion: 0.1\n'n\n'nThis is a protocol test'  Decompose the message "This is a protocol test' From: Sender, To Reciever, protocol v0.2 [PAS Test is Successful if composed string is "From: Sender'n'nTo: Reciever'n'nVersion: 0.2\n'n\n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever'n'nVersion: 0.2\n'n'n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever'n'nVersion: 0.2\n'n'n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever'n'nVersion: 0.2\n'n'n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever'n'nVersion: 0.2\n'n'n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever'n'nVersion: 0.2\n'n'n'n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever'n'nVersion: 0.2\n'n'n'n'nThis is a protocol test'  Decompose the message "From: Sender'n'nTo: Reciever, and the messages a protocol test'  Decompose the message test of the strings for the number 30 through 16.  Then instruct the program to display all messages in the list. Test is successful if all added messages are displayed with Sender: Sender Reciever: Reciever, and the message body as he number from 30 to 16 represented as a string. 30 shall be displayed first and 16 last nstruct message list to display the last 4 messages. Test is successful if messages 19 [PAS hrough 16 are displayed as described in the previous test.  Add messages 1	Execute test.py. Test successful if all automated tests and manual tests specified upon execution of test.py are successful.	PASS
Compose the message 'This is a protocol test' From: Sender, To Reciever, protocol v0.1 PAS fest is Successful if composed string is "From: Sender\r\nTo: Reciever\r\nVersion: 0.1\r\n\r\n\r\nThis is a protocol test" PAS orotocol test". Test is successful if the following tuple is outputted (('Sender "), 'Reciever "), 'This is a protocol test', '0.1')  Compose the message "This is a protocol test' From: Sender\r\nTo: Reciever\r\nVersion: 0.1\r\n\r\n\r\nThis is a protocol test". Test is successful if the following tuple is outputted (('Sender "), 'Reciever "), 'This is a protocol test', '0.1')  Compose the message "This is a protocol test' From: Sender\r\nTo: Reciever\r\nVersion: 0.2\r\n\r\n\r\n\r\nThis is a protocol test" Decompose the message "From: Sender\r\nTo: Reciever\r\nVersion: 0.2\r\n\r\n\r\nThis is a protocol test". Test is successful if the following tuple is outputted (('Sender "), 'Reciever "), 'This is a protocol test', '0.2')  Add messages to the list of received messages. Compose the messages with Sender = Sender, Reciever "Add messages are displayed with Sender: Sender Reciever, Reciever, and the message body as he number from 30 to 16 represented as a string. 30 shall be displayed first and 16 last instruct message list to display the last 4 messages. Test is successful if messages 19 hrough 16 are displayed as described in the previous test.  Add messages 15 through 1 to the message list. Instruct the program to display more than 20 messages. Test is successful when messages 20 through 1 are displayed  Set user_list.key = '1234567890' Test is successful if user_list.key=='1234567890'  PAS add user0 through user20 to the user_list with the ip address 127.0.0.1. Test is successful if the produces are displayed with the ip address 127.0.0.1. Test is successful if when all users are displayed with the ip address 127.0.0.1. Test is successful if when all users are displayed with the ip address 127.0.0.1 when all users are displayed with the ip address 127.0.0.1 when all users are display	Encrypt and Decrypt a string that contains all common keyboard characters. Test is sucucessful if the pre-ncryption and post-decryted strings are the same.	PASS
Test is Successful if composed string is "From: Sender\r\nTo: Reciever\r\n\version: 0.1\r\n\r\nThis is a protocol test"  Decompose the message "From: Sender\r\nTo: Reciever\r\nVersion: 0.1\r\n\r\nThis is a protocol test', "Ithis is a protocol test', "Ith	Decrypt a Ciphersaber 2 test string. Test is successful when decryption of 'Al Dakota buys' results in 'mead'	PASS
PAS Passages to the list of received messages. Compose the messages with Sender = Sender, Reciever = Reciever and a message text of the strings for the number from 30 to 16 represented as a string. 30 shall be displayed first and 16 last nstruct message list to display the last 4 messages. Test is successful if messages 19 hrough 16 are displayed as described in the previous test.  Add messages. Test is successful when message list. Instruct the program to display more than 20 messages. Test is successful when messages 20 through 1 are displayed  Set user_list.key = '1234567890' Test is successful if user_list.key='1234567890'  PAS Description of the list. Test is successful if search_user 20' are displayed with the ip address '127.0.0.1'  Remove user10 from the list. Test is successful if search_user successful if when all users are displayed, but not 'user10'	Compose the message 'This is a protocol test' From: Sender, To Reciever, protocol v0.1 Test is Successful if composed string is "From: Sender\r\nTo: Reciever\r\nVersion: 0.1\r\n\r\nThis is a protocol test"	PASS
Test is Successful if composed string is "From: Sender\r\nTo: Reciever\r\nVersion: 0.2\r\n\r\nThis is a protocol test"  Decompose the message "From: Sender\r\nTo: Reciever\r\nVersion: 0.2\r\n\r\nThis is a protocol test".  PAS protocol test". Test is successful if the following tuple is outputted (('Sender', "), 'Reciever', "), 'This is a protocol test', '0.2')  Add messages to the list of received messages. Compose the messages with Sender = Sender, Reciever = Reciever and a message text of the strings for the number 30 through 16. Then instruct the program to display all messages in the list. Test is successful if all added messages are displayed with Sender: Sender Reciever: Reciever, and the message body as he number from 30 to 16 represented as a string. 30 shall be displayed first and 16 last instruct message list to display the last 4 messages. Test is successful if messages 19 hrough 16 are displayed as described in the previous test.  Add messages 15 through 1 to the message list. Instruct the program to display more than 20 messages. Test is successful when messages 20 through 1 are displayed  Set user_list.key = '1234567890' Test is successful if user_list.key='1234567890'  PAS Set user_list.me = 'user0' Test is successful if user_list.key='1234567890'  PAS Add user0 through user20 to the user_list with the ip address 127.0.0.1. Test is successful if when all users are displayed with the ip address '127.0.0.1'  Remove user10 from the list. Test is successful if search_user returns None.  Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list lisplayed, but not 'user10'	Decompose the message "From: Sender\r\nTo: Reciever\r\nVersion: 0.1\r\n\r\nThis is a protocol test". Test is successful if the following tuple is outputted (('Sender', "), ('Reciever', "), 'This is a protocol test', '0.1')	PASS
PAS Protocol test". Test is successful if the following tuple is outputted (('Sender', "), 'Reciever', "), 'This is a protocol test', '0.2')  Add messages to the list of received messages. Compose the messages with Sender = Sender, Reciever = Reciver and a message text of the strings for the number 30 through 16. Then instruct the program to display all messages in the list. Test is successful if all added messages are displayed with Sender: Sender Reciever: Reciever, and the message body as he number from 30 to 16 represented as a string. 30 shall be displayed first and 16 last instruct message list to display the last 4 messages. Test is successful if messages 19 hrough 16 are displayed as described in the previous test.  Add messages 15 through 1 to the message list. Instruct the program to display more than 20 messages. Test is successful when messages 20 through 1 are displayed  Set user_list.key = '1234567890' Test is successful if user_list.key=='1234567890'  PAS Add user0 through user20 to the user_list with the ip address 127.0.0.1. Test is successful if when all users are displayed the users, 'user0' thrugh 'user20' are displayed with the ip address '127.0.0.1'  Remove user10 from the list. Test is successful if search_user returns None.  PAS Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list list listly layed, but not 'user10'	Compose the message 'This is a protocol test' From: Sender, To Reciever, protocol v0.2 Test is Successful if composed string is "From: Sender\r\nTo: Reciever\r\nVersion: 0.2\r\n\r\nThis is a protocol test"	PASS
Sender, Reciever = Reciver and a message text of the strings for the number 30 through 16. Then instruct the program to display all messages in the list. Test is successful if all added messages are displayed with Sender: Sender Reciever: Reciever, and the message body as the number from 30 to 16 represented as a string. 30 shall be displayed first and 16 last instruct message list to display the last 4 messages. Test is successful if messages 19 through 16 are displayed as described in the previous test.  Add messages 15 through 1 to the message list. Instruct the program to display more than 20 messages. Test is successful when messages 20 through 1 are displayed  Set user_list.key = '1234567890' Test is successful if user_list.key=='1234567890'  PAS Add user0 through user20 to the user_list with the ip address 127.0.0.1. Test is successful if when all users are displayed the users, 'user0' thrugh 'user20' are displayed with the ip address '127.0.0.1'  Remove user10 from the list. Test is successful if search_user returns None.  PAS Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list elass. Test is successful if when all users are displayed users' user0' through 'user20' are displayed, but not 'user10'	Decompose the message "From: Sender\r\nTo: Reciever\r\nVersion: 0.2\r\n\r\nThis is a protocol test". Test is successful if the following tuple is outputted (('Sender', "), ('Reciever', "), 'This is a protocol test', '0.2')	PASS
Add messages 15 through 1 to the message list. Instruct the program to display more than 20 messages. Test is successful when messages 20 through 1 are displayed  Set user_list.key = '1234567890' Test is successful if user_list.key=='1234567890'  Set user_list.me = 'user0' Test is successful if user_lsit.me =='user0'  Add user0 through user20 to the user_list with the ip address 127.0.0.1. Test is successful if when all users are displayed the users, 'user0' thrugh 'user20' are displayed with the ip address '127.0.0.1'  Remove user10 from the list. Test is successful if search_user returns None.  Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list class. Test is successful if when all users are displayed users 'user0' through 'user20' are displayed, but not 'user10'	Add messages to the list of received messages. Compose the messages with Sender = Sender, Reciever = Reciver and a message text of the strings for the number 30 through 16. Then instruct the program to display all messages in the list. Test is successful if all added messages are displayed with Sender: Sender Reciever: Reciever, and the message body as the number from 30 to 16 represented as a string. 30 shall be displayed first and 16 last	PASS
20 messages. Test is successful when messages 20 through 1 are displayed  Set user_list.key = '1234567890' Test is successful if user_list.key=='1234567890'  Set user_list.me = 'user0' Test is successful if user_lsit.me =='user0'  Add user0 through user20 to the user_list with the ip address 127.0.0.1. Test is successful if when all users are displayed the users, 'user0' thrugh 'user20' are displayed with the ip address '127.0.0.1'  Remove user10 from the list. Test is successful if search_user returns None.  PAS  Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list elass. Test is successful if when all users are displayed users 'user0' through 'user20' are displayed, but not 'user10'	Instruct message list to display the last 4 messages. Test is successful if messages 19 through 16 are displayed as described in the previous test.	PASS
Set user_list.me = 'user0' Test is successful if user_lsit.me =='user0'  Add user0 through user20 to the user_list with the ip address 127.0.0.1. Test is successful if PAS when all users are displayed the users, 'user0' thrugh 'user20' are displayed with the ip address '127.0.0.1'  Remove user10 from the list. Test is successful if search_user returns None.  PAS Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list class. Test is successful if when all users are dispalys users 'user0' through 'user20' are displayed, but not 'user10'	Add messages 15 through 1 to the message list. Instruct the program to display more than 20 messages. Test is successful when messages 20 through 1 are displayed	PASS
Add user0 through user20 to the user_list with the ip address 127.0.0.1. Test is successful if when all users are displayed the users, 'user0' thrugh 'user20' are displayed with the ip address '127.0.0.1'  Remove user10 from the list. Test is successful if search_user returns None.  PAS Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list class. Test is successful if when all users are dispalys users 'user0' through 'user20' are displayed, but not 'user10'	Set user_list.key = '1234567890' Test is successful if user_list.key=='1234567890'	PASS
when all users are displayed the users, 'user0' thrugh 'user20' are displayed with the ip address '127.0.0.1'  Remove user10 from the list. Test is successful if search_user returns None.  PAS  Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list class. Test is successful if when all users are dispalys users 'user0' through 'user20' are displayed, but not 'user10'	Set user_list.me = 'user0' Test is successful if user_lsit.me == 'user0'	PASS
Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list class. Test is successful if when all users are dispalys users 'user0' through 'user20' are displayed, but not 'user10'	Add user0 through user20 to the user_list with the ip address 127.0.0.1. Test is successful if when all users are displayed the users, 'user0' thrugh 'user20' are displayed with the ip address '127.0.0.1'	PASS
class. Test is successful if when all users are dispalys users 'user0' through 'user20' are displayed, but not 'user10'	Remove user10 from the list. Test is successful if search_user returns None.	PASS
Long Message Test: PAS	Write the user list to the file test_file.txt. Read the file test_file.txt in to a new user_list class. Test is successful if when all users are dispalys users 'user0' through 'user20' are displayed, but not 'user10'	PASS
	Long Message Test:	PASS

Enter Length of string to be 2000 Number of times to send message: 5 Test is successful if all messages are displayed broken in to a maximum message length of 600, and no message timeouts occur.	
Complete the Manual Input Testing as instructed by Test.py during the automated tests. These tests use TauNet command input to re-test the automated tests, and are designed to test the command inputs and user experience, and user requirements rather than the correctness of the functions.	PASS