

Software Testing Documentation

Prepared by CODEFELLAS2

for the project LINUX PASSWORD VAULT

**METU - Department of Computer
Engineering**

**CENG 492 Senior Design Project II
Spring 2015-2016**

Ali Can Oğul - 1746304

- [1. Introduction](#)
 - [1.1 Document Identifier](#)
 - [1.2 Scope](#)
 - [1.3 References](#)
 - [1.4 Level In The Overall Sequence](#)
 - [1.5 Test Classes And Overall Test Conditions](#)
- [2. Details For System Plan](#)
 - [2.1 Features To Be Tested](#)
 - [2.2 Features To Be Tested](#)
 - [2.3 Approach](#)
 - [2.4 Item Pass / Fail Criteria](#)
 - [2.5 Test Deliverables](#)
- [3. Test Management](#)
 - [3.1 Planned Activities and Tasks; Test Progression](#)
 - [3.2 Environment / Infrastructure](#)
- [4. Test Case Details](#)
 - [4.1 Introduction](#)
 - [4.2 Test Cases](#)
- [5. System Test Report Details](#)
 - [5.1 Overview Of Test Results](#)
 - [5.2 Detailed Test Results](#)
 - [5.3 Rationale For Decisions](#)
 - [5.4 . Conclusions And recommendations](#)

1. Introduction

1.1 Document Identifier

The project aims to create a service where all passwords can be stored securely, accessed by the authorized personnel only on a need-to-know basis.

This document is prepared by Codefellas2 with the purpose of testing the Linux Password Vault software. The document explains design of test cases and procedure with great detail so that any tester would be able to run and observe the outcomes. This document is the first version of Software Testing Documentation.

1.2 Scope

The purpose of this document is to provide the test cases of Linux Password Vault. The document provides the objective, scenario, expected outcomes and procedural requirements for each test case. Additionally, it contains a chart that shows the relation between test cases. The software will be tested using guidance of this document. Even though it embodies all the test cases specifically in detail, a little portion of the details is subject to change in test phase. In the first section, the background information about Linux Password Vault and the aim of this document are provided. The second section is dedicated to the details about system test plan, while the the third section gives management information about test. Finally,

last two chapters, form main part of the document, explaining all the test cases and giving the results.

1.3 References

[1] IEEE Standard for Software and System Test Documentation, IEEE Std 829 TM 2008

[2]SDD of Linux Password Vault, Software Design Description

1.4 Level In The Overall Sequence

- Integration testing: In this level, modules are combined together into different subsystems, which are then tested. The goal here is to see if the modules can be integrated properly. This testing activity can be considered testing the design.
- System testing: In this level, the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if the software meets its requirements.

- Acceptance testing: Acceptance Tests are the tests which will be performed by the stakeholders to validate the system to see if the system meet their needs or not. Acceptance Tests will be performed after the system test and performance tests.

In this document, only integration testing is followed in order to of ensure that the subsystems are able to work as a correlated system.

1.5 Test Classes And Overall Test Conditions

In the next sections of the document, many test cases are given a place for checking the software in terms of different perspectives.

2. Details For System Plan

2.1 Features To Be Tested

Projects functional properties will be tested. While testing environment variables and specific input parameters will be used to see the software's behavior and the related errors.

2.2 Features To Be Tested

During testing process capability (access control) will not be tested because it's not finished exactly yet.

2.3 Approach

Blackbox testing which examines the functionality of an application without peering into its internal structures or workings is determined to be used based on our knowledge of internal operation and implementation of Linux Password Vault.

2.4 Item Pass / Fail Criteria

For each test there are input-output pairs which must be satisfied by related test cases. The test cases which result in unexpected outputs shall be considered as fail. This approach shall be applied only to test cases with black box method.

2.5 Test Deliverables

This document contains test procedure specifications that explain the steps for executing all test cases given in the following sections and the results of each test case. It is prepared based on IEEE Standard for Software and System Test Documentation.

3. Test Management

3.1 Planned Activities and Tasks; Test Progression

The test process will start with analysis and inspection of the SDD in order to understand the requirements and design cases and code in order to find the error prone parts of the code in testing purposes. Then, necessary test cases will be determined and for better understanding, cases will be divided into groups according to their objectives. After that, the expected outputs for each test case will be decided and described. Finally, inputs for each test case will be determined.

In the final step, the results of each test will be gathered in a test results table. This will enable us to see what should be done in further steps of the software's development. The testing of this project will be done manually except performance testing.

3.2 Environment / Infrastructure

Hardware Needs: The devices which will form the hardware components of the system shall have network access. Therefore, modem, WAN/LAN, Ethernet crosscable is considered as hardware needs.

Software Needs: A Linux Operating System with terminal is required.

4. Test Case Details

4.1 Introduction

This section provides the detailed explanation for each test case accompanied by the its inputs, outcomes, environmental and procedural requirements along with the dependencies among test cases. This section includes the information for all test cases we run on the project.

For each test case, there are seven fields; test case identifier, objective, inputs, outcomes, environmental needs, special procedural requirements, intercase dependencies. Test case id is unique for each test case and is used for identifying test cases.

4.2 Test Cases

Test case identifier	Test1
Objective	Logging in
Inputs	username, password
Outcome	User is logged in
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	There are no intercase dependencies for this test case.

Test case identifier	Test2
Objective	Logging out
Inputs	quit command
Outcome	User is logged out
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	Test1

Test case identifier	Test3
Objective	Adding user
Inputs	new username and password
Outcome	New user is added
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	Test1

Test case identifier	Test4
Objective	Storing password
Inputs	new domain and password
Outcome	New domain and its password are stored.
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	Test1

Test case identifier	Test5
Objective	Getting Password
Inputs	domain name
Outcome	Wanted domain password is retrieved.
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	Test1, Test4

Test case identifier	Test6
Objective	Allow another user to access stored domain password
Inputs	domain name and allowed user's username
Outcome	User gained access.
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	Test1, Test4

Test case identifier	Test7
Objective	Revoke allowed user to access stored domain password
Inputs	domain name and allowed user's username
Outcome	Access is revoked.
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	Test1, Test4, Test6

Test case identifier	Test8
Objective	Adding a user to a user group.
Inputs	username, groupname
Outcome	User is added and gained access to group's domains.
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	Test1

Test case identifier	Test9
Objective	Removing a user from a user group.
Inputs	username, groupname
Outcome	User is removed and lost access to group's domains.
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	Test1

Test case identifier	Test10
Objective	Allow all group members to access a domain.
Inputs	groupname, domain name
Outcome	All group members gained access.
Environmental needs	Linux operating system which supports Ruby and MySQL
Special procedural requirements	There are no special procedural requirements.
Intercase dependencies	Test1

5. System Test Report Details

5.1 Overview Of Test Results

In this section of the document, test cases that are explained in the previous section are concluded with their results according to pass/fail criteria that is also defined in the document. Results of the all test cases are given in following part.

5.2 Detailed Test Results

Test1	PASS
Test2	PASS
Test3	PASS
Test4	PASS
Test5	PASS
Test6	PASS
Test7	PASS
Test8	PASS
Test9	PASS
Test10	PASS

5.3 Rationale For Decisions

Test cases were formed in order to see the behavior of the code in prone parameters and inputs. The test cases were tested number of times and behavior of the system was observed using our predefined evaluation metrics.

5.4 . Conclusions And recommendations

In the testing period, 10 different test cases are executed. All of the test are about functionality testing about the project.