Automation framework development for enhancing security functionalities of Oracle's planning and prediction application

Submitted By

Harsha Punjabi

16MCEI17

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Automation framework development for enhancing security functionalities of Oracle's planning and prediction application

Major Project

Submitted in fulfillment of the requirements

for the degree of

Master of Technology in Computer Science & Engineering

(*Information & Network Security*)

Submitted By

Harsha Punjabi

(16MCEI17)

Guided By

Prof. Pooja Shah

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Certificate

This is to certify that the major project entitled **"Automation framework development for enhancing security functionalities of Oracle's planning and prediction application"** submitted by **Harsha Punjabi (Roll No: 16MCEI17)**, towards the fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering(Information & Network Security) of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Pooja Shah
Guide & Assistant Professor,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sharada Valiveti
Coordinator M.Tech - CE (INS),
CE Department,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor & Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Alka Mahajan
Director,
Institute of Technology,
Nirma University,
Ahmedabad.

Statement of Originality

I, **Harsha Punjabi**, **16MCEI17**, give undertaking that the Major Project entitled " **Automation framework development for enhancing security functionalities of Oracle's planning and prediction application**" submitted by me, towards the fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering***(Information & Network Security)* of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof .Pooja Shah

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to Prof. Pooja Shah, Assistant Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank Dr. Sanjay Garg, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to Dr. Alka Mahajan, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation she has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Harsha Punjabi

16MCEI17

Abstract

It is important in software industry to deliver the right product at the right time to customer with all the quality checks and also maintaining the security posture of the application is one of the greatest priorities. Manual testing becomes difficult and time consuming, if one tries various input combinations. Moreover, it is costly to manually analyze all results. This project is motivated by the drawbacks of manual regression testing and how to overcome those drawbacks using automated tools for ensuring maintaining the overall security of the software.

The Project "Automation framework development for enhancing security functionalities of Oracle's planning and prediction application" includes work on the OSSA automation and other automation for the Retail Predictive Application Server (RPAS) web client as well as server. Also, development of Automation Libraries used frequently in different applications will be done using Perl scripting language & Synergy automation tool during this project.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The biggest challenge to all software industry in today's world is to maintain and improve the efficiency,security and quality of the software product continuously as security is an important property of any software. As on a large scale, organizations require their data to be secured in the cloud as well as on premises, the concept of Software Security Assurance is ever growing in it's importance.

Test automation is one of the main solutions to improve the quality and assure the security of software development process in many ways. Tests which are automated run with a comparatively faster speed and much more easily because of the functional modules re usability feature within different tests.

If the software projects are not managed properly and do not follow a proper predefined procedure, then there are high chances for the software to fail.The similar thing applies to test automation.Testing engineers know that how important it is to follow a specific flow to develop software automation framework in order to make a software work according to predefined standards promised to customers.

By testing all the major security functionalities of a software, we can assure that the security standards are met and in accordance.By testing we can find out the security bugs and through patch we can have them gone thus retaining the overall security posture.Thus testing plays a major role in SSA.

Chapter 2

Literature survey

This chapter includes the survey done in order to understand the security fundamentals and automation flow and it's importance.The following points were drawn out of the literature surveyed.

Agile development is a better and much flexible way to develop software applications. Testing is the most integral and important part of any software development. Agile method almost follows the same approach as the traditional software testing, but there are certain differences in these to approaches.Quality has to be assured and infused during the entire the entire product life cycle.To help deliver the product effectively and in a time efficient manner, the selection of the automation testing tool plays a crucial role [11].

The increase in the importance of test automation in software engineering can be seen by the increasing number of companies that have invested in various automation testing tools. This huge money is invested in order to prevent the defects that comes up during the product development process. A core and important activity for agile methodologies is test automation and is the key for speeding up the process of quality assurance. All the challenges & observations of a testing project team which is completely new to the agile practices and also the test automation which can be done using open source testing tools are described in detail.[2].

The current state of art in software security assurance is of a great topic of discussion.The different variety of technologies used by the government,corporate industry and

academia to specify, acquire, produce, assess, and deploy a software that can be said to be secure to an extent are discussed and described[4].

Integrating the software security activities in an agile based Software Development Life cycle is a big task. As previously the companies adapted a waterfall approach in which security activities used to fit in well, it was a great challenge to fit in the agile methodology also. This paper describes the changes required to be performed in an organization who is moving from waterfall to agile methodologies.The importance of synchronizing the security tests with the agile rhythm of sprints is described in depth in this paper[1].

Chapter 3

RPAS

3.1 Introduction

RPAS stands for Retail Predictive Application Server.It is a platform which can be configured and is used for developing highly scalable planning and forecasting based solutions with proven scalability. RPAS platform has the following capabilities[8]

- powerful multidimensional database
- online and batch processing
- slice and dice configurable UI
- calculation engine
- user security functions
- utility functions and other powerful functions
- can be deployed on a variety of hardware

3.2 RPAS Platform Components

The main RPAS Platform components are:

1. RPAS Server

   RPAS server consists of two main components :

- RPAS Domain
- Domain Daemon

2. RPAS Client
3. Configuration Tools

RPAS Platform concepts has been depicted in the Figure 3.1.

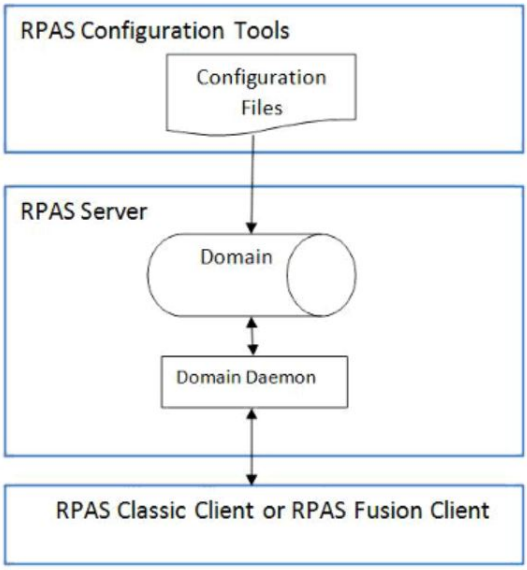


Figure 3.1: RPAS Patform Concepts

3.3 RPAS Server

- Main Component of RPAS server are[9]:

  1. RPAS Domain
  2. RPAS Domain Daemon

- RPAS domain are the server side files and directories that houses data used by RPAS database.
- RPAS domain uses Multi-dimensional database to store data used by RPAS.
- RPAS server also stores the "Platform Code base" that provides a set of utilities to interact with the domain.

3.3.1 RPAS Domain Daemon

RPAS domain daemon is a middle ware between RPAS client and domain. It can be defined as follows:

- Process that enables communication channel between RPAS Client and RPAS domain
- It is a server side utility that will wait for request from Client on specific port
- After request is received,server process is started to which client connects to

3.3.2 RPAS Features

This section introduces to the following RPAS concepts[5]:

- Multidimensional data storage
- Dimensions
- Calculation Engine
- Measures
- Domains and Workbooks

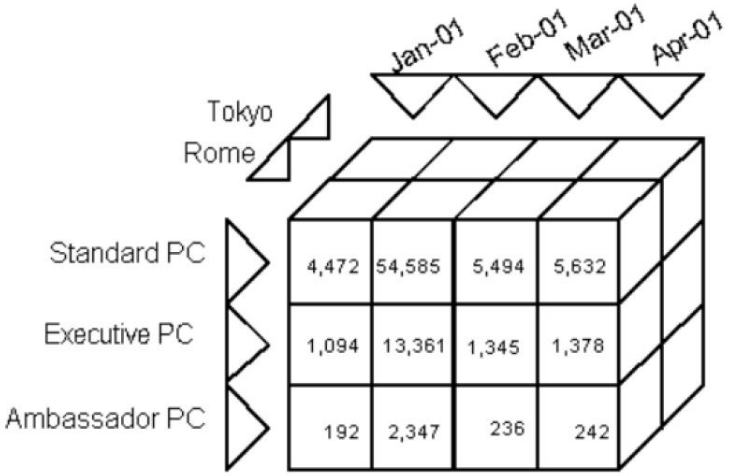


Figure 3.2: Mutlidimensional cube

- Multidimensionality:-In RPAS, the storage and representation of data is in a multi-dimensional framework.The data is represented as a multidimensional array,where each of the values is contained in a cell which can be accessed through multiple indexes.

  Multidimensional database systems achieve performance levels above the relational database systems are a complementary technology to entity relational systems.

  Multidimensionality provides the following features:

  - Easy determination of number of dimensions and positions
  - Aggregating and spreading the data
  - Increases the speed of data analysis and retrieval as it need not search individual records
  - Data can be sliced and diced easily

  A Multidimensional cube has been depicted in the Figure 8.9.

- Calculation Engine:- Using calculation engine the following tasks could be done:

  - Aggregation
  - Spreading
  - Expression Evaluation

  A measure's base intersection is combination of dimensions which defines the lowest level at which data can be stored or held for the measure. Aggregation maintains the values of measures above their base intersection.RPAS needs to change the underlying data values at base intersection for the measure in order to preserve the integrity of data so that when the meaures are again aggregated, they reflect a changed value at aggregated level.For this spreading is used to change the values at base intersection.

- Dimensions:-Dimensions can be defined as the qualities of an item where the item can be a product, location, or time value.The components of a dimension actually define the structure and roll up within the dimension.

  Dimensions defines the relationship between the different levels of dimensions.The

dimensions that are set up at user business and used by the merchandising solutions are reflected by these dimensions.

Many alternative dimensions are supported by the application which can provide different roll ups and also help the user of the application to analyze the data from a various perspectives.

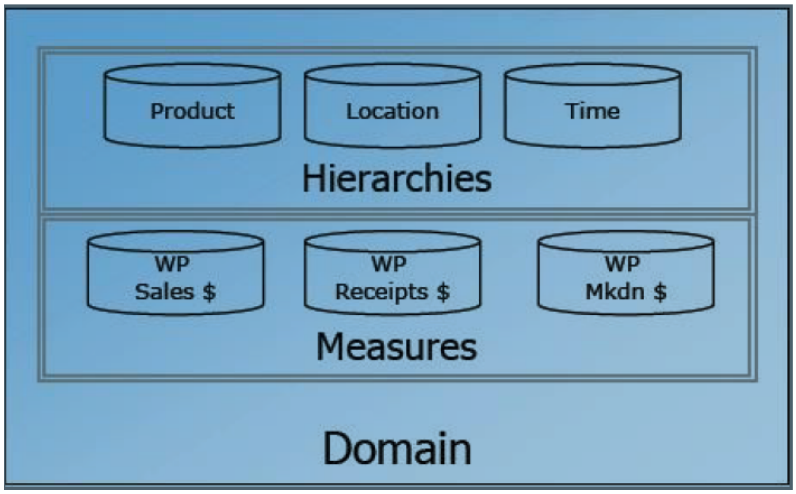


Figure 3.3: RPAS Concepts

- Measures:- Measures are the values or the measurements that are recorded.Rule sets are used to define the relationships between various measures in the application. The base intersection of a measure defines or decides the dimensionality of a measure.

  Measure name configuration is possible and they are named through a predefined convention.

- Domains and Workbooks:- RPAS stores teh data in a cache .This cache is:

  - persistent
  - multidimensional
  - can handle huge volumes of data

  The cache which is the central repository for data storage is called a domain. This central repository includes the detailed definitions of the metadata for the all the avaliable solutions & provides a single point of update.

  Workbook is a personal data repository used by users.When user wants to access an RPAS solution, the user interacts with that solution through a workbook.[6]

A workbook does not contain the whole data but instead contains a part of i.e a subset of the metadata and data.The scope is restricted by the access rights given to a user for that domain.

Workbooks can be built through a wizard process and are stored on the central RPAS server.In a workbook the data will remain independent of the domain.

3.4 RPAS Client

RPAS Client can be defined as:

- RPAS Client allows user to access and manipulate data in the domain of RPAS Server through a windows GUI
- RPAS Client allows user to build a personalized scaled down version of domain using user-defined data limitations. These personalized copies offers various views of data stored in the domain database.

A user will access the domain via the RPAS Client. When in the domain, a user will access a workbook wizard specific to the workbook template they want to build. After the user makes their selections, RPAS will build the workbook. The building of a workbook includes building the structure of the workbook, loaded the current data from the domain, running the calculations for non-loaded measures and data is aggregated to meet the needs of the view in the workbook. After the workbook is built, it is available to a user to make updates and changes. A user is required to save the workbook in order for it to be used later. When saving a workbook, the changes are still isolated to the individual workspace (i.e. workbook). In order for other users to see the updates and for new workbooks to contain that updated data, the workbook must be committed back to the domain.

3.5 RPAS Configuration Tools

RPAS Configuration tools can be defined as :

- Provides a flexible mean to build and configure RPAS based applications with customer specific parameters
- Provides a user-friendly,streamlined interface to utilize RPAS functionality

Chapter 4

OSSA

4.1 Software Security Assurance

"Software Security Assurance (SSA) is the process of ensuring that software is designed to operate at a level of security that is consistent with the potential harm that could result from the loss, inaccuracy, alteration, unavailability, or misuse of the data and resources that it uses, controls, and protects".[12]
The whole process of SSA first starts by requirement identification and then classifying the information which is to be used by the software.
Security testing mainly focuses on finding out software weaknesses or vulnerabilities and recognizing the various unexpected or unidentified situations because of which the software could fail in such a way that will violate the security conformance.[4]

4.1.1 OSSA

Oracle Software Security Assurance (OSSA) is the procedure that Oracle follows for incorporating security into every phase of the product development life cycle. i.e the design, build, testing, and maintenance phases of it's products. Oracle aims to ensure that the products are meeting their security requirements which is satisfactory for customers. The phases process that is followed by Oracle and the various phases in which security is incorporated is shown in the figure 4.1.

Functional security testing is a type of testing which mainly focuses on testing the sofware's security functionalities.Functional security testing are executed by the product's Quality assurance team along with normal product testing cycle. During this testing phase, Quality Assurance engineers verify conformance of implemented security features

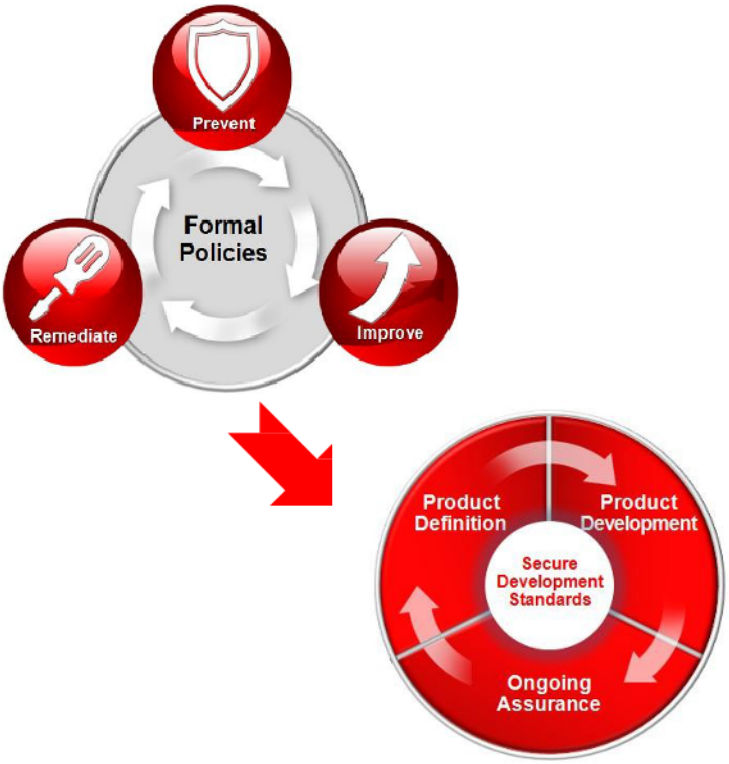


Figure 4.1: OSSA phases

to what had been previously agreed upon in the functional specifications during the checklist reviews process.

The following figure 4.2 shows what all tests are done in a software to assure it's overall security.

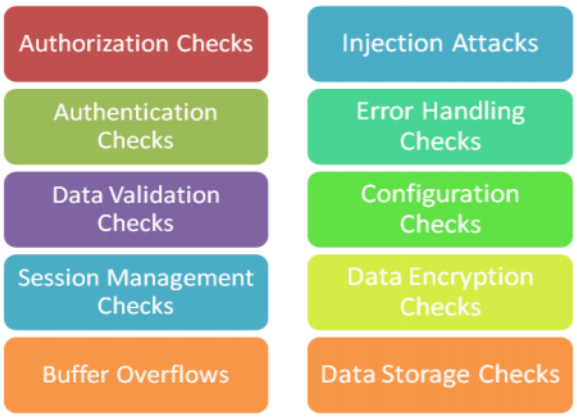


Figure 4.2: Seurity tests

4.2 Security in RPAS

The following are the security areas in RPAS:

- Operating System Level Security:- This includes security considerations regarding operating system file permissions, account creation, and folder permissions.
- User and Group Management:- This includes granting rights to a group and to a user that belongs to a particular group.Also whether a user needs to inherited the default group rights has to be specified.
- Managing user accounts:- Includes User Account Lockouts,Password history, Password expiration.
- Authorization:- Includes Workbook Security,Measure Level Security,Position Level Security
- Setting Proper Resource Limits:- Includes Workbook Template Limits Views,Max Domain Session Limit View,Max User Session Limit View,Dimension Modification Rights View
- Auditing:- The RPAS_LOG_LEVEL environment variable establishes the minimum logging level used by the RPAS server.
- Managing Sensitive Data:- Sensitive data like passwords must not be revealed in any form and should always be protected.
- Domain Daemon IP Filtering and Redundancy:- RPAS supports the concept of blocking some IP addresses in a multi-homed server from being used by the Domain Daemon, thereby limiting the security exposure to external attacks.
- Secure Socket Layer:- SSL is a protocol for securing the network connectionsa and is used by the applocation for providing communication that is secure between the Client and server processes.
- Configuration Security:- All admin tasks are predefined in xml files and put under the config folder of the domain with AdminTasks.xml as the suffix of the file names. These files should be protected by changing their UNIX file permissions to read-only to only the RPAS UNIX administration account.

Chapter 5

System Implementation

This sections describes the overall architecture of the system and give an insight to the OJET UI that the product uses.
RPASCE is a configurable cloud-engineered platform with a proven scalability for developing multidimensional forecasting and planning-based solutions with an enhanced user experience.
The RPASCE Client is the web-based client for the RPASCE platform developed using the latest Oracle JavaScript Extension Toolkit (Oracle JET). It delivers an enhanced user experience for the RPASCE platform.The flow diagram of the system is shown in figure 5.1.

RPASCE Concepts[3]:

- RPASCE : It is a platform providing foundation for running solutions used for retail planning. These solutions are provided with common interface containing wizards, templates, workbooks and batch processes.
- RPASCE Solution : This is an application which run on top of RPASCE and it provides solutions for various retail problems such financial planning or forecasting demand.
- RPASCE Domain : A collection of server side directories containing directories and procedure to execute a specific RPASCE solution.

Users access an RPASCE solution through the RPASCE client,a web-based client.

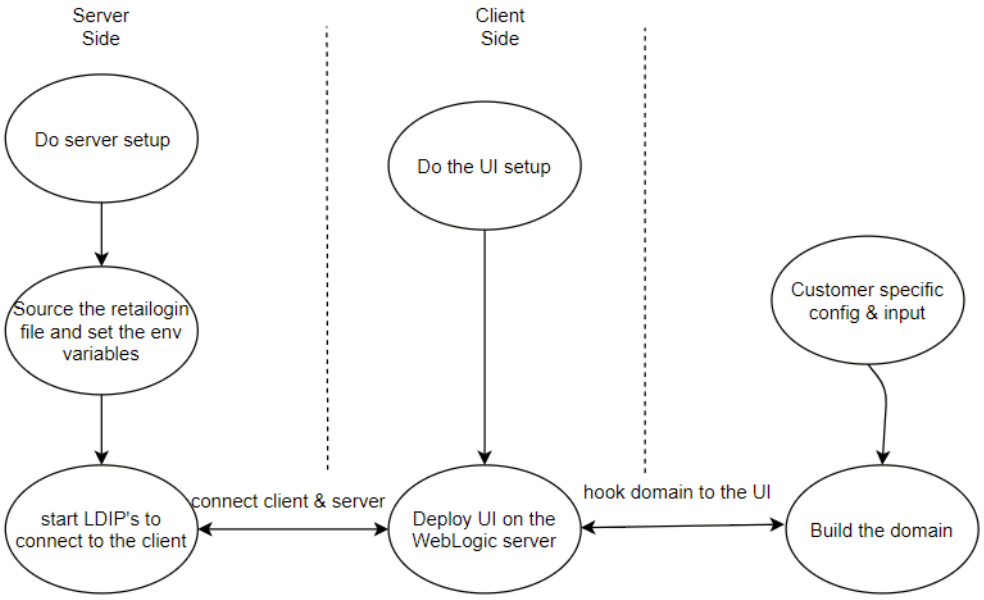


Figure 5.1: System Implementation

Chapter 6

Methodology

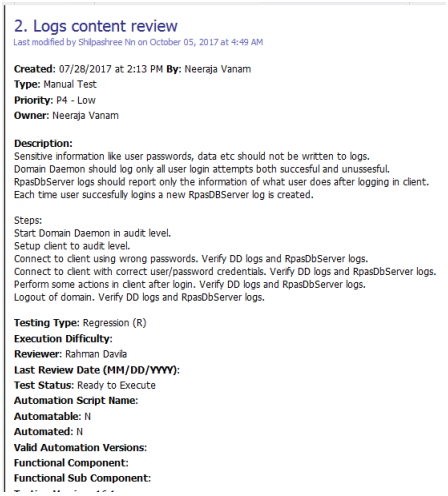
This sections describes the methodology used to design the Testing Framework for the RPAS Platform.[10]

6.1 Test Case Design

Test cases are designed using Oracle Test Manager (OTM). It contain the following sub sections for each test case:

- Details
- Description
- Test Steps
- Details: The details give all the auxiliary information for the test case. It
- Description: The description part explains the overview of the test case under consideration. The description part includes the purpose of the test, pre-requisite for the tests, overview of the steps, impacted tables in all the applications involved in the test case .
- Test Steps: The design steps consists of the Step number, the description of the steps and the expected outcome when each step is executed.

  The following figure 6.1 shows the sub sections of a test case



2. Logs content review

Last modified by Shilpashree Nn on October 05, 2017 at 4:49 AM

Created: 07/28/2017 at 2:13 PM **By:** Neeraja Vanam
Type: Manual Test
Priority: P4 - Low
Owner: Neeraja Vanam

Description:
Sensitive information like user passwords, data etc should not be written to logs.
Domain Daemon should log only all user login attempts both succesful and unussesful.
RpasDbServer logs should report only the information of what user does after logging in client.
Each time user succesfully logins a new RpasDBServer log is created.

Steps:
Start Domain Daemon in audit level.
Setup client to audit level.
Connect to client using wrong passwords. Verify DD logs and RpasDbServer logs.
Connect to client with correct user/password credentials. Verify DD logs and RpasDbServer logs.
Perform some actions in client after login. Verify DD logs and RpasDbServer logs.
Logout of domain. Verify DD logs and RpasDbServer logs.

Testing Type: Regression (R)
Execution Difficulty:
Reviewer: Rahman Davla
Last Review Date (MM/DD/YYYY):
Test Status: Ready to Execute
Automation Script Name:
Automatable: N
Automated: N
Valid Automation Versions:
Functional Component:
Functional Sub Component:

Figure 6.1: Test case subsections

6.2 Test Data

Test data is the data which has been specifically identified for use in tests, typically of a computer program. Some data may be used in a confirmatory way, typically to verify that a given set of input to a given function produces some expected result.[13]

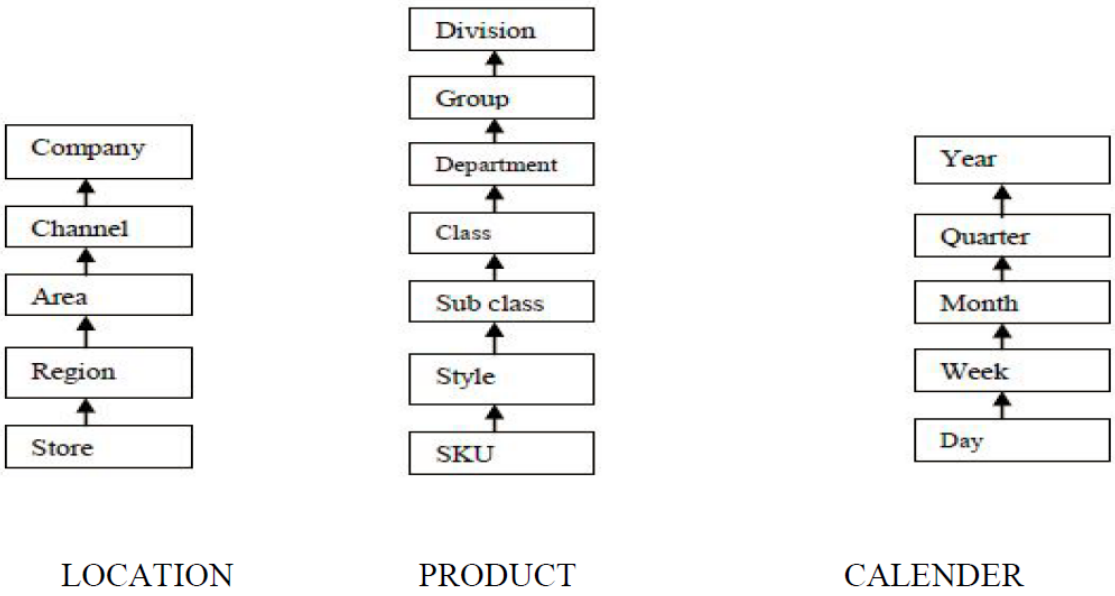


Figure 6.2: Dimension Hierarchy

Test data can be produced by the legacy team by using help of a program or function that aids the tester to produce the test data. The data can be stored for re-use or also be forgotten if not needed. This data used is created in Retail Predictive Application Server includes the data for product,calender and location hierarchies as shown in figure 6.2.

Chapter 7

Server Side Automation

Server side automation involves automating the security functionalities on the server side i.e back end. This is done using automated scripts written in Perl.The entire flow of this server side automation is described in a flow Design that explains the way in which the testing strategy proceeds. Requirements and ER are checked for duplication, if there is no duplication then a test case is written and is tested manually as well as automated. Depending on its execution, the defect is passed or failed. Flow Design changes based on the needs of the organization. Figure 7.1 shows the sample flow design.

The flow of the whole Test Automation Framework is depicted in the Figure 7.2. The attribues.ini file contains all the required environment variables. The library contains the moduleRBT.pm which has all the commonly used modules. The main driver script uses this attributes.ini file and library.
An individual script contains the test cases for a particular functionality. Each testcase is depicted by a function, which uses the library modules. Each individual script is called by the main Driver script. On the basis of the run status, the pass or error logs are generated. The logs are managed by a reporting mechanism.

7.1 Perl Scripting

PERL SCRIPTING is for major automation of functionalities. This mainly include Perl command and also the shell scripting.

- Efficient because scripts are reusable and easy to understand and execute.
- As this does not require any tool it can be run at the back end and also saves time.

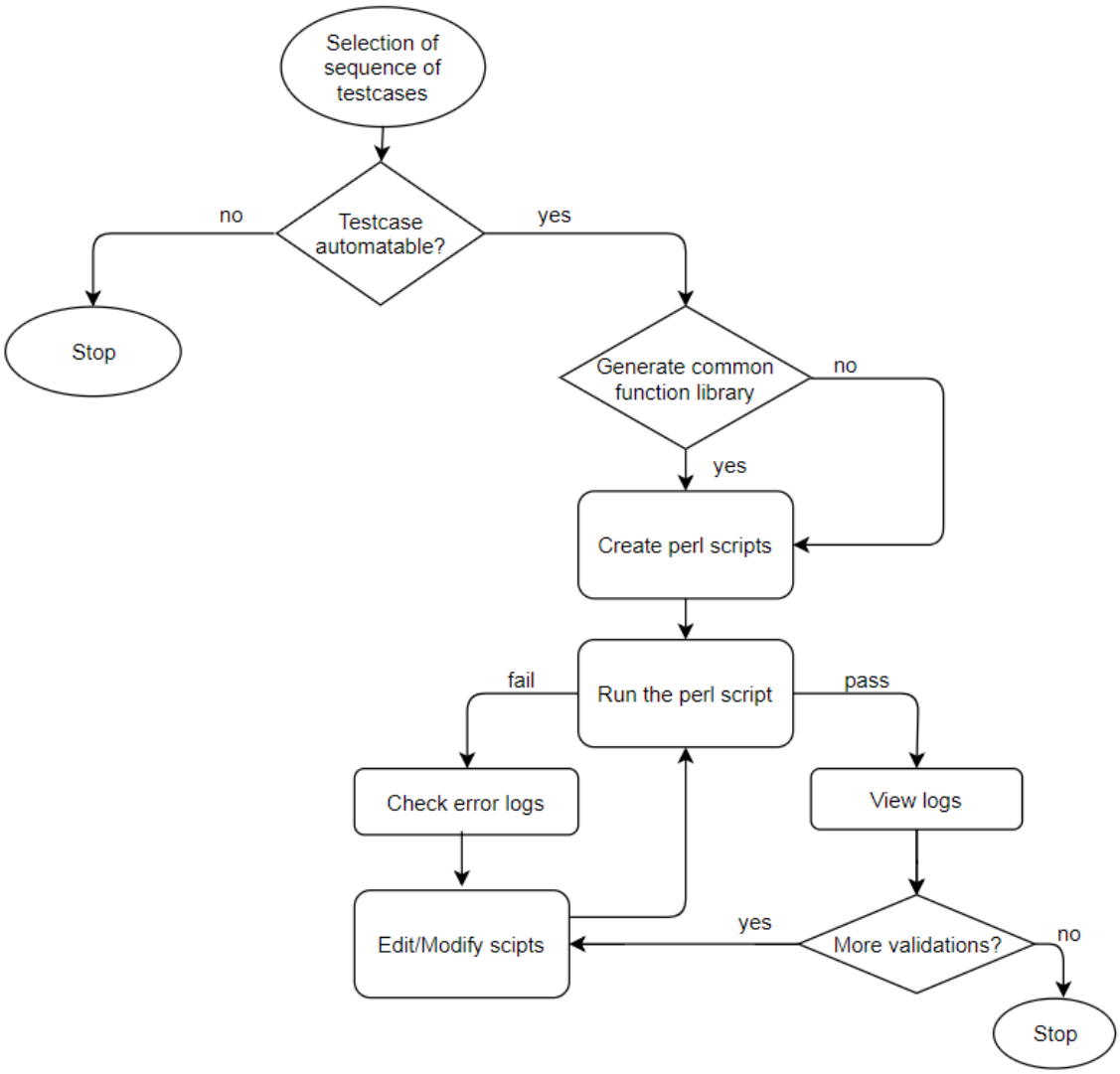


Figure 7.1: Test case automation flow

- Results are usually displayed on the screen and the log files are created for each test case.
- In case of test case fails error log will be generated and it has detailed information of command which has failed and the reason for the failure.

All the test case are combined into one single script which calls all the functionalities, that script file is known as testDriver.pl and common functions will be saved in separate file known as ModuleRBT.pm which contains all the common functions used in the script.

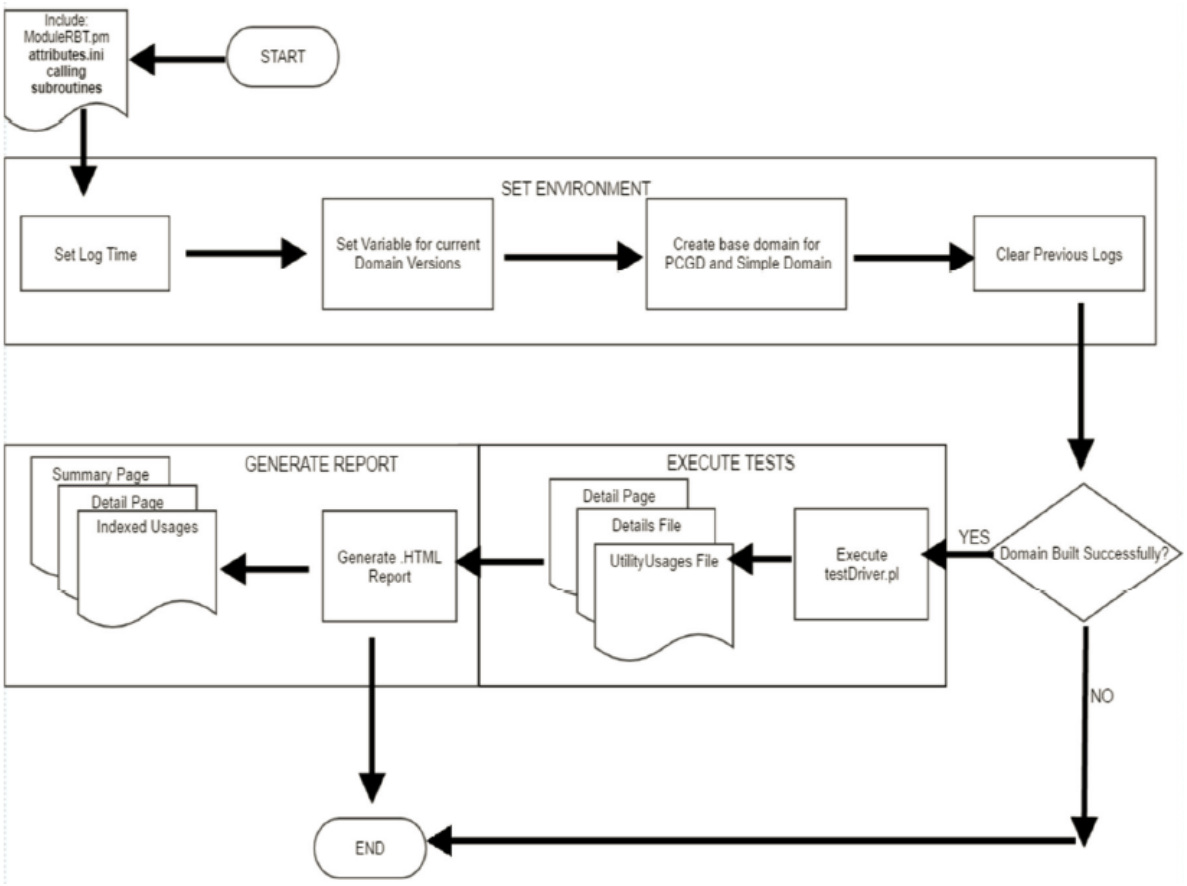


Figure 7.2: Automation Flow

7.2 Development of Automation Library

Automation libraries consists of various scripts functions dealing with the several functionality of the application which is used frequently.

Automation Libraries includes the functions which we can be used in automating all the applications of RPAS Platform These functions are made only for common functionality of different applications and utilities.

For EG : sub Login( ). Login needs to be done for each and every application domain so we have created the function of "Login".So we can reuse the same.We need to just add a Login function with the required arguments(username,password,port and domain) to login into any domain.

The perl module which contains all these developed functions is ModuleRBT.pm and is shown in the following figure

ModuleRBT.pm

```
1  package ModuleRBT;
2  use List::Util;
3  use Config::IniFiles;
4  use Cwd;
5  use Cwd qw(abs_path);
6  use File::Basename;
7  use Text::CSV;
8  use Capture::Tiny ':all';
9
10
11 BEGIN {
12     use Exporter();
13
14     @ISA = qw(Exporter);
15     @EXPORT =
16       qw($DETAILS $BASEDOMAIN $DOMAIN $DOMAINNAME $DOMAINCOPY $SIMPLEDOMAIN $DOMAINHOME $DOMAINCOPYHOME $SIMF
       $OUTPUT $CHECKRESULT $RSC_PATH $WALLET_PATH $PORT $walletDir $LOG_DIR $OUTPUT $INPUT $DEFAULTPORT &EDGE
       &EDGE_FuzzingTestSetup &checkFileExists &RSC_callscript &fileCompare &callRSC &FuzzingTestSetup &OATTes
       &edit_RSC_connconfig &edit_RSC_runconfig &check_value  &parseDetails &writeHeader &writeLogin &check_re
       &LoginSecure &array_diff &writeErrLog &writeLog &iniRead &iniWrite &verifyLog &matchit &verifyCommand &
       &initLog &countTest &customErrorLog  &customLog &check_fileContent &countLinesInFile &countWordsInFile
       &replaceStringInFile &parse_csv_header &parse_csv_column &buildWB &testSetup &simpleSetup &logTime &che
17
18 }
19
20 ###Initialize variables###
21 my $cwd = cwd();
22 my $cfg1 = Config::IniFiles->new( -file => "$cwd/att.ini" );
23
24 ##Logging##
25 $LOG_DIR = $cfg1->val( 'attr', 'log.dir');
26 our $LOG_DIR = abs_path($LOG_DIR);
27 $resultLogFolder = "Reporting";
28 $resultsFile = $cfg1->val( 'attr', 'results.file' );
29 $detailsFile = $cfg1->val( 'attr', 'details.file' );
```

Figure 7.3: ModuleRBT.pm

7.3 Sample test cases

The following section shows the sample test cases which are run to assure that the software complies to the defined security standards.

1. DD & RpasDbserver logs location

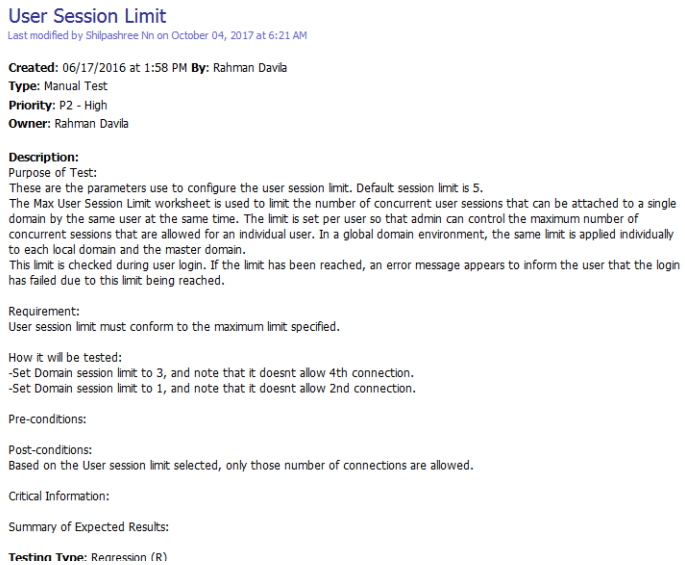
Last modified by Rahman Davila on July 08, 2015 at 7:33 PM

Created: 07/08/2015 at 7:33 PM **By**: Rahman Davila
Type: Manual Test
Priority: P3 - Medium
Owner: Rahman Davila

Description:
DomainDaemon and RpasDbServer Logs location.
DomainDaemon logs are stored in the directory where you launch DomainDaemon. RpasDbServer logs are stored in the authenticated user's directory inside the domain.

Steps:
Create PCGD domain, create users.
Start DomainDaemon in audit level, verify log is created from where it is started.
Start client in audit level, verify rpasDbServer log created under domainhome/users/specific user directory.
Stop daemon and close client.

Figure 7.4: Test case 1



User Session Limit

Last modified by Shilpashree Nn on October 04, 2017 at 6:21 AM

Created: 06/17/2016 at 1:58 PM **By**: Rahman Davila
Type: Manual Test
Priority: P2 - High
Owner: Rahman Davila

Description:
Purpose of Test:
These are the parameters use to configure the user session limit. Default session limit is 5.
The Max User Session Limit worksheet is used to limit the number of concurrent user sessions that can be attached to a single domain by the same user at the same time. The limit is set per user so that admin can control the maximum number of concurrent sessions that are allowed for an individual user. In a global domain environment, the same limit is applied individually to each local domain and the master domain.
This limit is checked during user login. If the limit has been reached, an error message appears to inform the user that the login has failed due to this limit being reached.

Requirement:
User session limit must conform to the maximum limit specified.

How it will be tested:
-Set Domain session limit to 3, and note that it doesnt allow 4th connection.
-Set Domain session limit to 1, and note that it doesnt allow 2nd connection.

Pre-conditions:

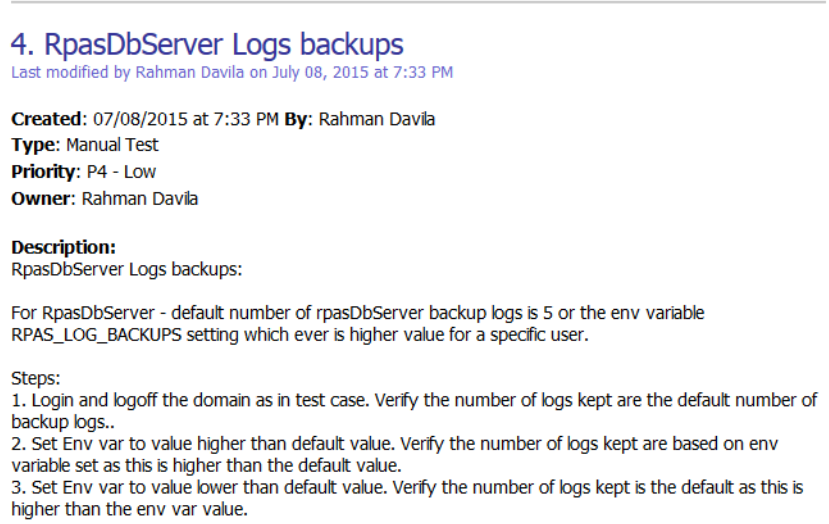
Post-conditions:
Based on the User session limit selected, only those number of connections are allowed.

Critical Information:

Summary of Expected Results:

Testing Type: Regression (R)

Figure 7.5: Test case 2



4. RpasDbServer Logs backups

Last modified by Rahman Davila on July 08, 2015 at 7:33 PM

Created: 07/08/2015 at 7:33 PM **By**: Rahman Davila
Type: Manual Test
Priority: P4 - Low
Owner: Rahman Davila

Description:
RpasDbServer Logs backups:

For RpasDbServer - default number of rpasDbServer backup logs is 5 or the env variable RPAS_LOG_BACKUPS setting which ever is higher value for a specific user.

Steps:
1. Login and logoff the domain as in test case. Verify the number of logs kept are the default number of backup logs..
2. Set Env var to value higher than default value. Verify the number of logs kept are based on env variable set as this is higher than the default value.
3. Set Env var to value lower than default value. Verify the number of logs kept is the default as this is higher than the env var value.

Figure 7.6: Test case 3

02_01 Password Aging - A

Last modified by Mohima Chaudhuri on August 07, 2016 at 3:32 PM

Created: 07/08/2015 at 7:32 PM **By**: Rahman Davila
Type: Manual Test
Priority: P3 - Medium
Owner: Rahman Davila

Description:
Password expiration may be enabled for a domain.
The domain administrator selects a number of days after which passwords expire. When a user logs in, if the configured number of days have passed since this user entered a new password, the system requires a new password to be entered.
Password expiration can be enabled through the domainprop utility using the -expirePassword flag.

Figure 7.7: Test case 4

03_01 Auto disable of accounts - A

Last modified by Mohima Chaudhuri on August 07, 2016 at 3:32 PM

Created: 07/08/2015 at 7:32 PM **By**: Rahman Davila
Type: Manual Test
Priority: P3 - Medium
Owner: Rahman Davila

Description:
Account Lockout may be enabled for a domain.
The domain administrator selects a number of failed logon attempts after which the User account will be marked as locked out. The account will remain locked out until the administrator re-enables it.
Account Lockout can be enabled through the domainprop utility using the -lockAccount flag.

Figure 7.8: Test case 5

Chapter 8

Client Side Automation

Client side automation is the UI automation which involves automating the security functionalities on the client side i.e front end. This is done using an Oracle internal automation tool called Synergy.
The entire flow of UI automation is described in a flow design that explains the way in which the testing strategy proceeds. Firstly the feaibility of the test is decided. If the automation of those set of tests is feasible then keywords are designed accordingly which can be used in the scripts to enhance re usability.Once the tests are automated, they are run all together to analyze the results. Figure 8.1 shows the sample flow design.

8.1 Synergy tool

Synergy is the software tool which is used to automate the Test-scripts. Synergy is developed using the Languages like JAVA, JAVASCRIPT and PYTHON.

Synergy provides UNIFIED AUTOMATION PLATFORM that provides :

- Easy and fast way of adding new automation plug-ins for new technologies
- Common test execution and recording architecture across all automation plug-ins
- Common automation integrated development environment (IDE) across all automation plug-ins

8.1.1 Synergy Test Suites

A Test Suite is a collection of tests that validate whether a software program exhibit a specific behaviour and any supporting constructs/information necessary for the execution

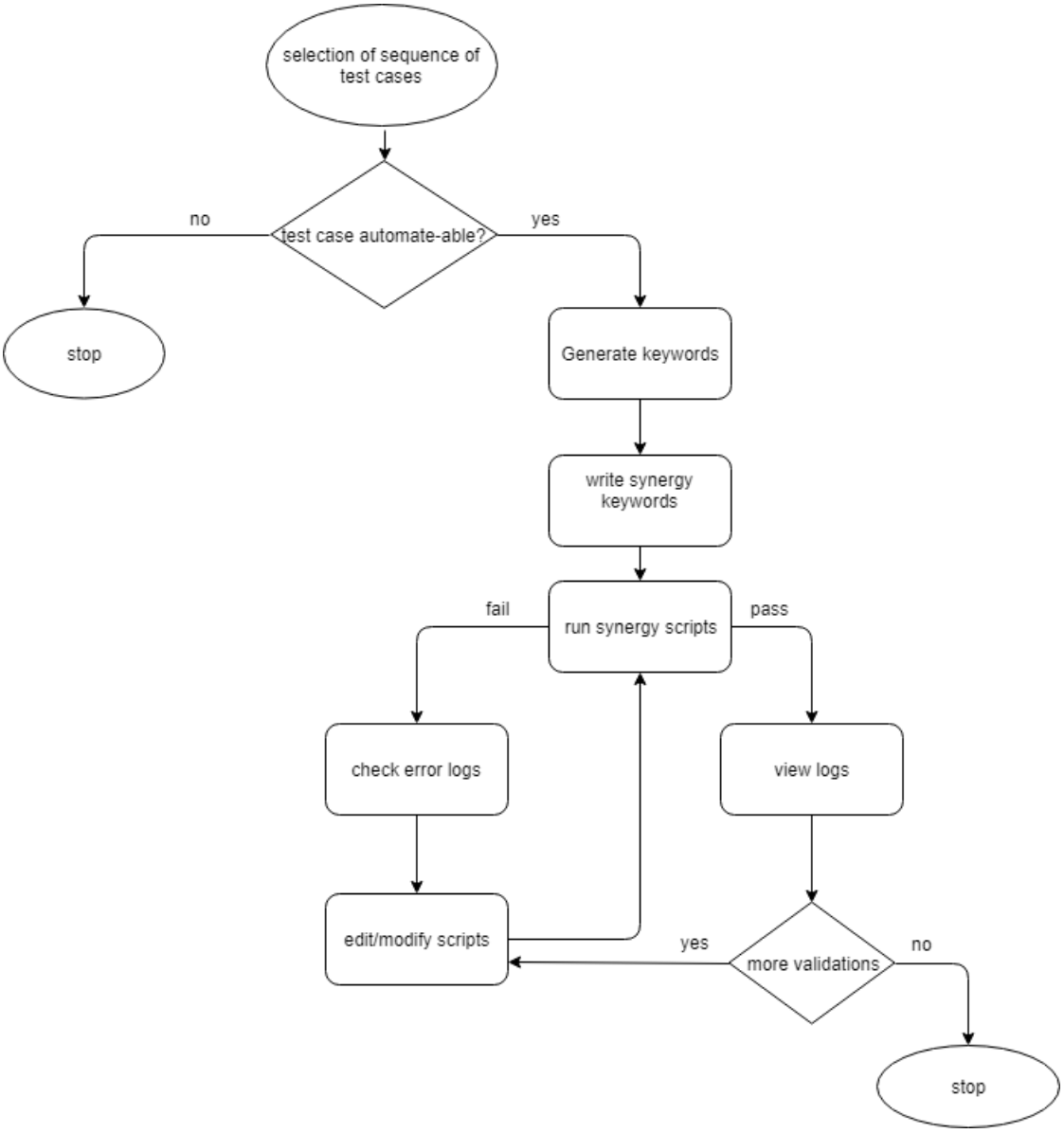


Figure 8.1: Test automation flow

of contained tests Synergy Test Suite contain a following component :

- Test Cases
- Folders
- User Keywords
- Resources Links
- Global Variables
- Configuration Settings

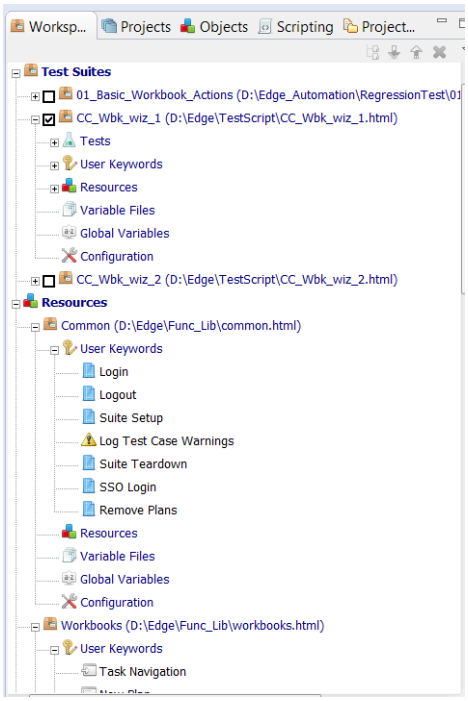


Figure 8.2: Synergy Workspace

8.1.2 Test Suite Creation

A test suite is created by selecting the Automation→Test Suite wizard accessible from the New button in the Synergy main toolbar or under File→New.

Once you have selected the new Test Suite wizard, you will be prompted for a test suite name and location.

8.1.3 Test Suite Configuration

Each test suite has configuration information associated with it. The configuration information may be accessed by double clicking on the Configuration of your test suite in the Workspace tab.

Under the Execution Settings section, you can configure test suite setup and teardown keywords.

8.1.4 Suite Setup Keyword

A test suite setup keyword can be used to perform any operations that may be needed before the execution of tests such as environment setup, database connection setup, data

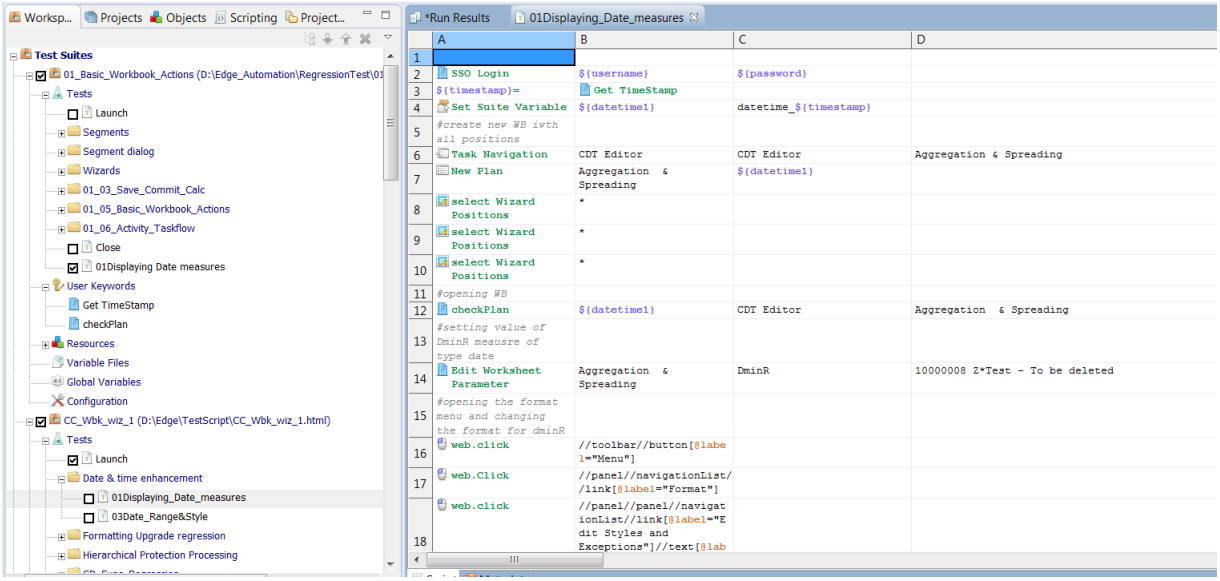


Figure 8.3: Synergy Test Suites

aggregation, etc.

Following are important points regarding suite setup:-

- A test suite setup keyword runs only once when the test suite is executed, before any tests contained in the test suite have run.
- If a test suite has a suite setup, the setup is executed before any of the test cases.
- If the suite setup passes, test execution continues as normal.
- If the suite setup fails, none of the test cases are executed.
- Suite setups are often used for setting up the test environment.
- Since tests are not run if the suite setup fails, suite setups are extremely useful to verify that the environment is in a state in which the tests can be executed successfully.

8.1.5 TestSuite Teardown

A test suite teardown keyword can be used to perform any cleanup or error recovery operations that may be needed after the execution of tests has completed. A test suite teardown keyword runs only once when the test suite has finished executing regardless of whether the tests contained in the test suite have run successfully.

If the testsuite includes Suite Teardown at the end,whatever steps are mentioned in Teardown are executed at the end of the Testsuite regardless of the success or failure of the test cases.

If the suite teardown fails, all tests in the suite are marked failed afterwards. Suite teardowns are commonly used for cleaning up the test environment after execution.

To ensure all teardown tasks are performed, all the keywords used in the teardown are executed even if some of them fail

8.2 Automation Library

Automation libraries contain the lowest-level keywords, often called library keywords, which actually interact with the AUT(Application Under Test).

All test cases always make the use of keywords from some library, often through higher-

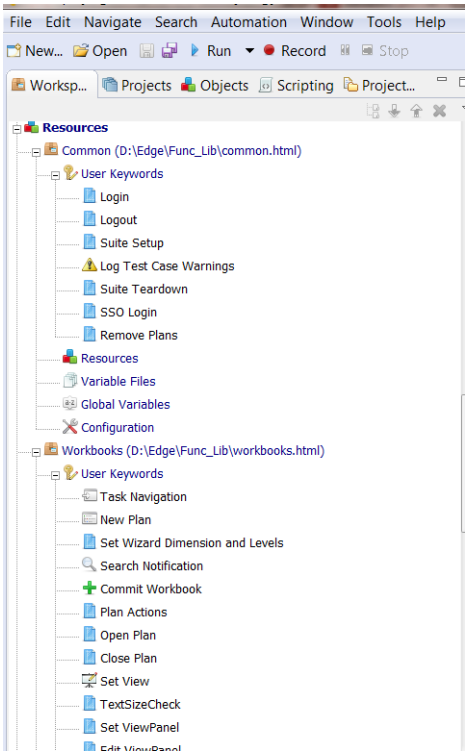


Figure 8.4: Synergy User defined keywords

level user keywords. Before any keyword provided by Synergy can be used in your test cases, you must explicitly import the library that provides that keyword into your test suite configuration.

Automation libraries can be imported by clicking on the Import library button which will

then present you with an import wizard associated with that library. For most libraries, you can simply click Finish in the wizard in order to import the library. For some libraries, however, you may have to provide additional parameters (such as database connection information, for example) on various import wizard screens.

A few of the various keywords used in the existing system are shown in figure ??.

8.3 Sample test cases

The following section shows the sample OSSA UI test cases which are run to assure that the UI complies to the defined security standards.

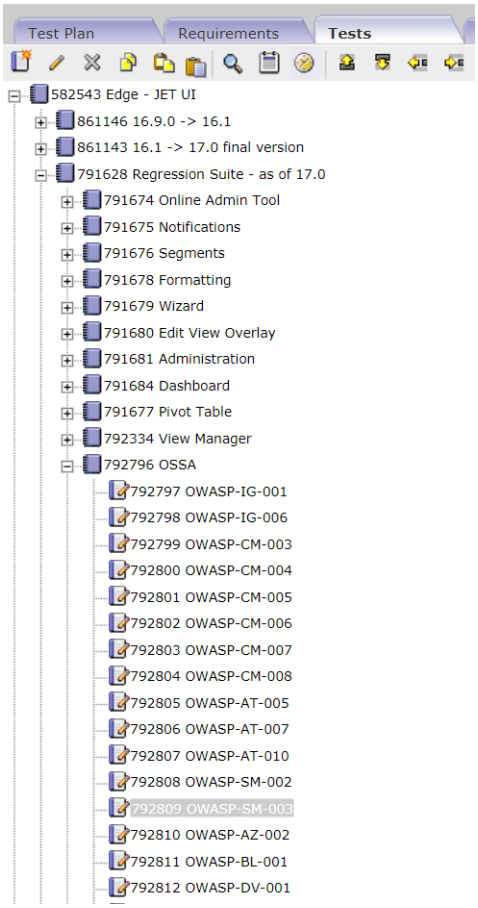


Figure 8.5: Test case folder structure

Test 792798 Run Test Edit Test Print e-Mail

OWASP-IG-006

Last modified by vinay subramani on May 17, 2017 at 4:10 PM

Created: 11/07/2016 at 3:39 AM **By**: Pradeep Jain
Type: Manual Test
Priority: P2 - High
Owner: Pradeep Jain

Description:
This test case verifies that the error messages shown to the user are generic and contain no sensitive information.

Testing Type: Regression (R)
Execution Difficulty:
Reviewer:
Last Review Date (MM/DD/YYYY):
Test Status: New
Automation Script Name:
Automatable: N
Automated: N
Valid Automation Versions:
Functional Component:
Functional Sub Component:
Testing Version: 16.1
Product: Oracle Retail Atlantis - 5255
SE Release Type:
Current Patch Version:
Patch Testing Version:
Testing Categories:
HPQC ID:

Test Steps Add/Edit

#	ACTION	EXPECTED RESULT	COMMENT	ATTACHMENTS
1	Login to JET UI as an admin user over a secure protocol.	User logged in and navigated to the homepage.		
2	Launch the wizard for Merch Plan Target	User will be thrown an error message "Please select at least one item". The message		

Figure 8.6: Test case 1

Test 792800 Run Test Edit Test Print e-Mail

OWASP-CM-004

Last modified by vinay subramani on May 17, 2017 at 4:10 PM

Created: 11/07/2016 at 3:39 AM **By**: Pradeep Jain
Type: Manual Test
Priority: P2 - High
Owner: Pradeep Jain

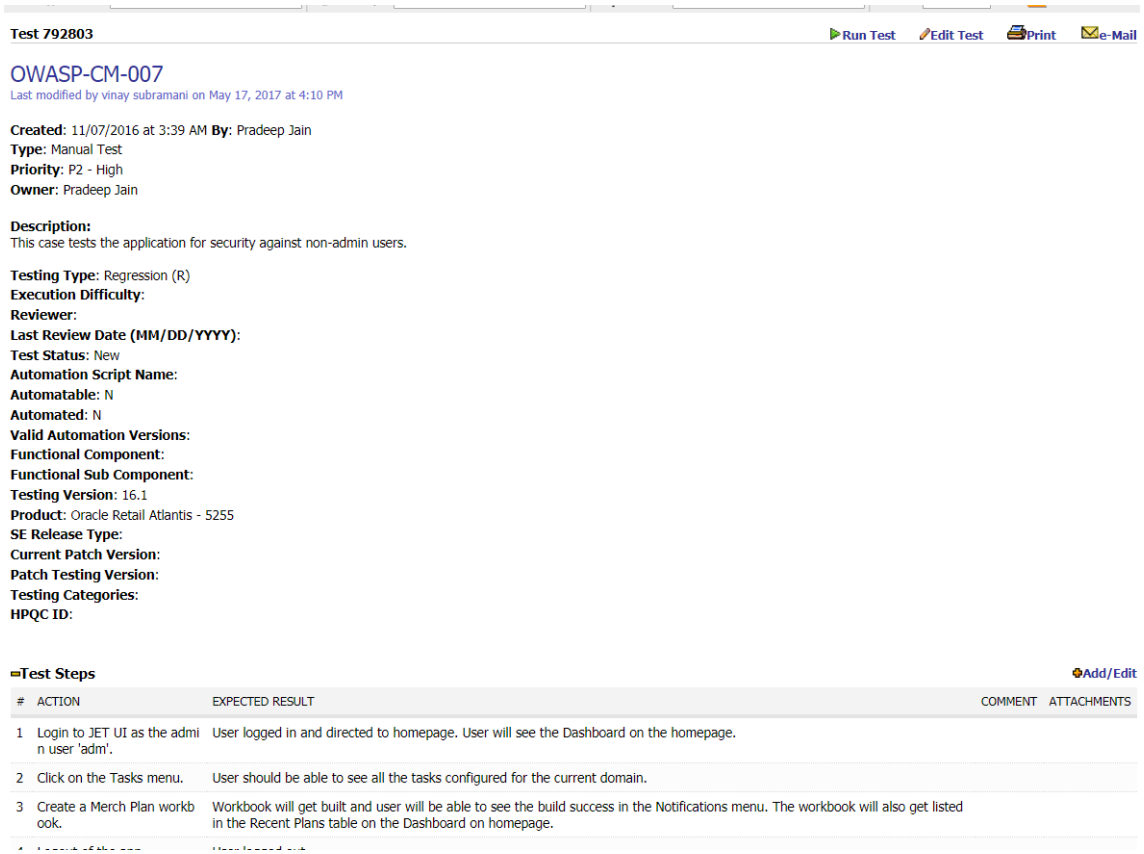
Description:
This case tests that the application has been configured in a secure manner.

Testing Type: Regression (R)
Execution Difficulty:
Reviewer:
Last Review Date (MM/DD/YYYY):
Test Status: New
Automation Script Name:
Automatable: N
Automated: N
Valid Automation Versions:
Functional Component:
Functional Sub Component:
Testing Version: 16.1
Product: Oracle Retail Atlantis - 5255
SE Release Type:
Current Patch Version:
Patch Testing Version:
Testing Categories:
HPQC ID:

Test Steps Add/Edit

#	ACTION	EXPECTED RESULT	COMMENT	ATTACHMENTS
1	Deploy JET UI on the WebLogic domain. Verify the directories and files created with the deployment.	There should be no unwanted/unnecessary files in the deployment directory.		
2	Open the log files and verify the content.	There should be no sensitive info in the files.		
3	Verify the presence of previous days' log files on the disk.	The previous logs should be renamed to, for example, edge.log.<date stamp>. This prevents a single file from getting long. It also makes discarding old log files convenient.		

Figure 8.7: Test case 2



Test 792803

Run Test | Edit Test | Print | e-Mail

OWASP-CM-007

Last modified by vinay subramani on May 17, 2017 at 4:10 PM

Created: 11/07/2016 at 3:39 AM **By**: Pradeep Jain
Type: Manual Test
Priority: P2 - High
Owner: Pradeep Jain

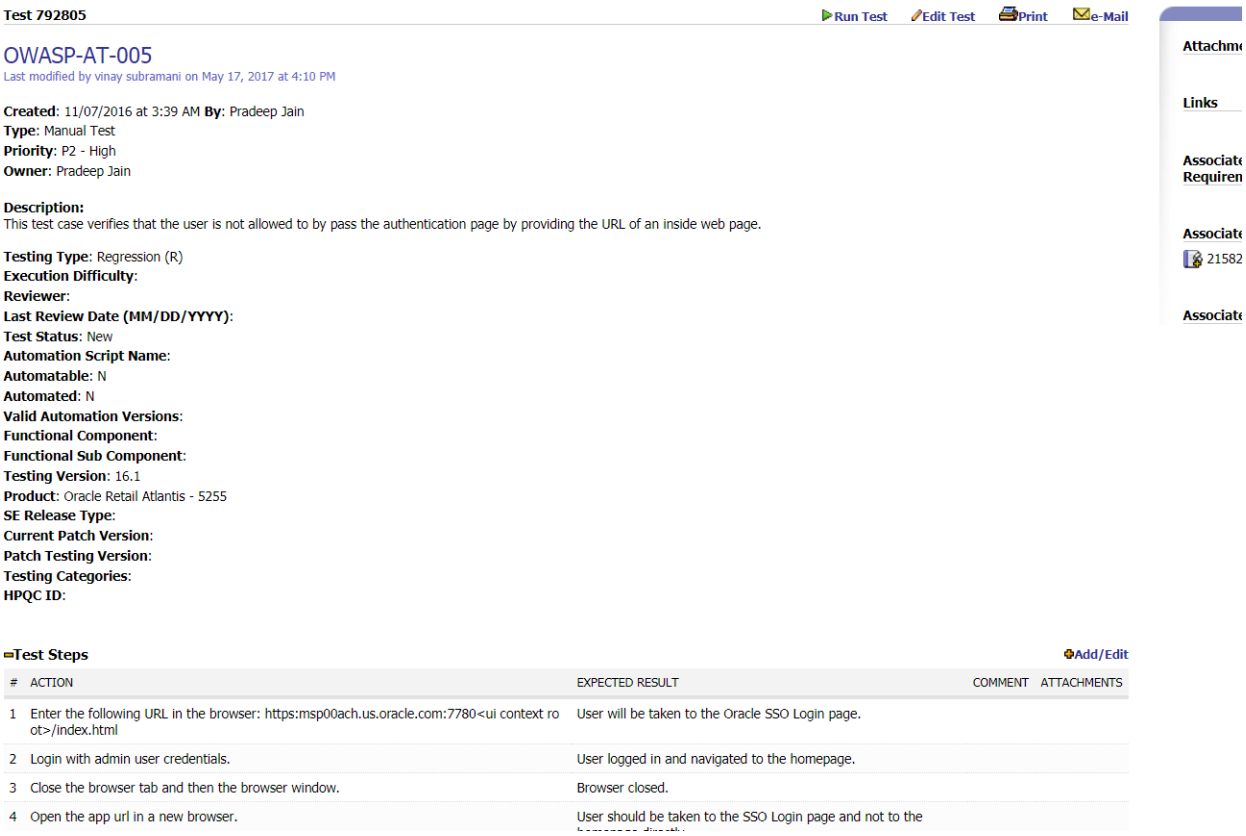
Description:
This case tests the application for security against non-admin users.

Testing Type: Regression (R)
Execution Difficulty:
Reviewer:
Last Review Date (MM/DD/YYYY):
Test Status: New
Automation Script Name:
Automatable: N
Automated: N
Valid Automation Versions:
Functional Component:
Functional Sub Component:
Testing Version: 16.1
Product: Oracle Retail Atlantis - 5255
SE Release Type:
Current Patch Version:
Patch Testing Version:
Testing Categories:
HPQC ID:

Test Steps — Add/Edit

#	ACTION	EXPECTED RESULT	COMMENT	ATTACHMENTS
1	Login to JET UI as the admin user 'adm'.	User logged in and directed to homepage. User will see the Dashboard on the homepage.		
2	Click on the Tasks menu.	User should be able to see all the tasks configured for the current domain.		
3	Create a Merch Plan workbook.	Workbook will get built and user will be able to see the build success in the Notifications menu. The workbook will also get listed in the Recent Plans table on the Dashboard on homepage.		

Figure 8.8: Test case 3



Test 792805

Run Test | Edit Test | Print | e-Mail

OWASP-AT-005

Last modified by vinay subramani on May 17, 2017 at 4:10 PM

Created: 11/07/2016 at 3:39 AM **By**: Pradeep Jain
Type: Manual Test
Priority: P2 - High
Owner: Pradeep Jain

Description:
This test case verifies that the user is not allowed to by pass the authentication page by providing the URL of an inside web page.

Testing Type: Regression (R)
Execution Difficulty:
Reviewer:
Last Review Date (MM/DD/YYYY):
Test Status: New
Automation Script Name:
Automatable: N
Automated: N
Valid Automation Versions:
Functional Component:
Functional Sub Component:
Testing Version: 16.1
Product: Oracle Retail Atlantis - 5255
SE Release Type:
Current Patch Version:
Patch Testing Version:
Testing Categories:
HPQC ID:

Attachm
Links
Associat
Requiren
Associat
21582
Associat

Test Steps — Add/Edit

#	ACTION	EXPECTED RESULT	COMMENT	ATTACHMENTS
1	Enter the following URL in the browser: https:msp00ach.us.oracle.com:7780<ui context root>/index.html	User will be taken to the Oracle SSO Login page.		
2	Login with admin user credentials.	User logged in and navigated to the homepage.		
3	Close the browser tab and then the browser window.	Browser closed.		
4	Open the app url in a new browser.	User should be taken to the SSO Login page and not to the		

Figure 8.9: Test case 4

Chapter 9

Wallet Creation Automation

This chapter will explain the work that has been done to automate the tedious wallet creation process using the orapki utility to reduce the manual effort and save time to a much greater extent.

9.1 Orapki Utility

The orapki utility is basically used to manage public key infrastructure (PKI) elements which include wallets and certificate revocation lists. It is provided on the command line so that the incorporation of the tasks it performs can be done into scripts. By finding a way to incorporate the PKI elements and their management into a script makes automation of the tasks of maintaining a PKI much easier.

9.1.1 Authentication Process

The procedure that is followed in an Oracle environment for authentication is described as follows[7]:

1. The Oracle database server is sent a connection request by the client.
2. A handshake is performed by the Secure Sockets Layer. During this handshake, the server authenticates itself to the client and then the client and server mutually decides upon which cipher suite they will be using.
3. After the handshake is completed successfully, the user requests a database access.

4. User gets authenticated by the Oracle database server. The authentication server uses a non-SSL authentication method like Kerberos or RADIUS.

5. Once the authentication server's validation is performed, the Oracle database server will grant access as well as authorization to the user, and after then only the user can securely access the database by using SSL.

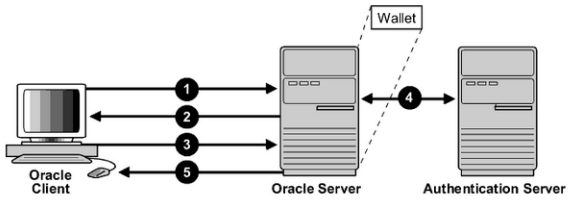


Figure 9.1: Authentication Process

9.1.2 Wallets

A wallet is a container that stores authentication and signing credentials, including private keys, certificates, and trusted certificates SSL needs [].
Oracle Wallet Manager is used by the security administrators to manage security credentials on the server. It is used by the wallet owners to manage security credentials on clients. The Oracle Wallet Manager is used to do the following:

- Generate a public-private key pair and create a certificate request
- Store a user certificate that matches with the private key
- Configure trusted certificates

9.1.3 The createSSLWallets.sh script

The createSSLWallets.sh script is a script that automates the entire wallet creation and management process. It is an interactive script that prompts for user inputs. The following is the list of input it asks for:

- The directory paths :

  This is the location where we want the wallets to get created. Under this directory, a directory named "rpaswallets" gets created which will store all wallets under itself. To make sure that some old stale wallets do not mix up with the new wallets, the

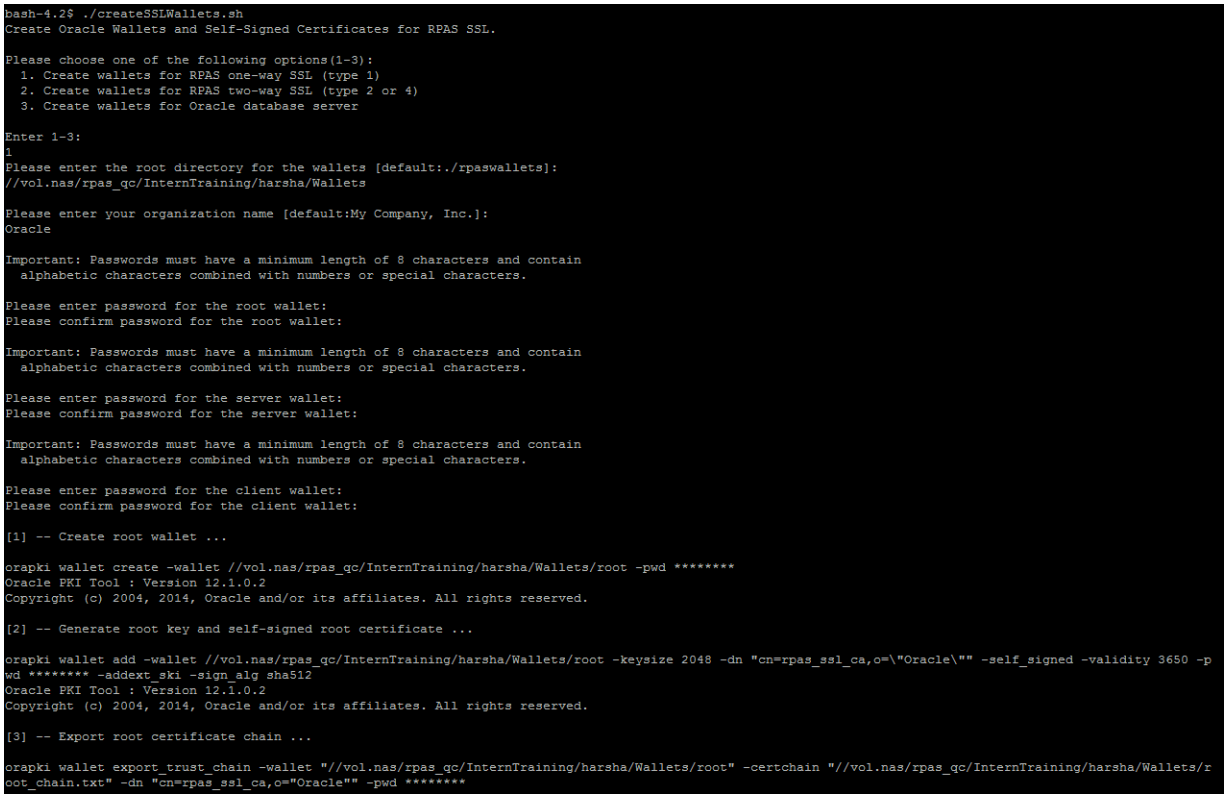


Figure 9.2: The create wallet script

script checks whether the wallets directory is already existing or not. If yes, then user should first remove that directory and then proceed. If no input is given wallets get created in the current working directory by default.

- Organization name :

  The user is asked to enter an organization name that is used as part of the DN (Distinguished Name) of the certificate identities.

- Passwords for root wallet, server wallet, & client wallet :

  Passwords entered are validated twice and also blank passwords are rejected.

- Output :

  The output of the scripts are the steps that are performed in order to create the wallets and certificates. A total of 17 steps is performed along with the orapki command line for each step with password masked out is displayed. Also any error message from orapki utility is also displayed and if error is encountered the scripts terminates immediately. After the script gets executed successfully, the locations of the root wallet, server wallet, and client wallet are displayed.

Chapter 10

Result and Analysis

This chapter will include the results/output of various configurations, function libraries and modules. The significance of the results/outputs will also be discussed along with the deviation from the actual results. .

10.1 Test Reports

Test reports effectively communicate our findings and conclusions to the application's stakeholders. Test Report is often generated to make the stakeholders aware of the findings. Test reporting involves the analysis of the information available to support conclusions, recommendations, and decisions about how to guide the project forward.

Test Reports are generated on the overall testing effort, which is generated on a weekly basis and as well as at the end of executing each test script. Weekly test status reports give the customer, an insight into the Total Number of Test Cases written, the number of test cases that were automated, the number of test cases that were manually executed and it accounts the failures along with the percentage of automation. Figure 4.2 shows the test status report.

Test Report generated after the execution of each test script contains detailed analysis of each step in the test script. The status of each step in the test script is explicitly explained. It also gives the reason for its failure, if any. The report gives the name of the step that is executed, the duration of execution, its time stamp, the time taken for response, the result of execution whether it fails or not and its summary. The summary section gives a detailed description of the data used for execution of each particular step, comments and warnings, if present are displayed. Failures are also reported.

10.2 Server Side Reporting

The main testDriver.pl script is run from the back-end server using "perl testDriver.pl linux attribues.ini" command from the location where the file exists.After the run the output for every testcase is generated. This maybe pass or fail result.

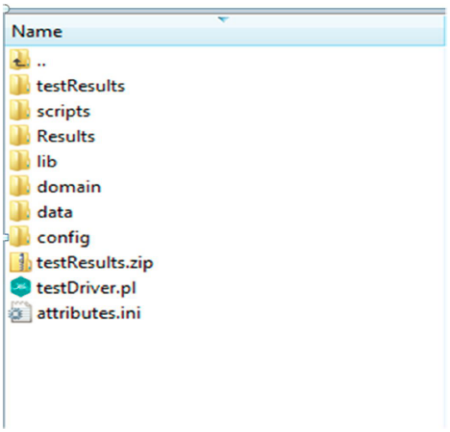


Figure 10.1: Directory Structure of Automation Framework

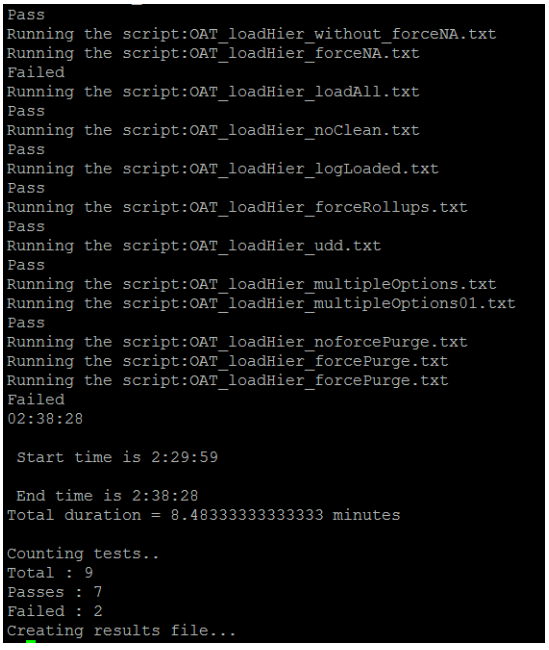


Figure 10.2: After the Run

All the test cases are written in a particular directory structure which is shown in Figure 10.1 .This directory structure includes the following:

Verifying purge data is deleted	PASSED	DataLifeSpan01.log
Verifying purge data is deleted	PASSED	DataLifeSpan02.log
Domain Session Limit	PASSED	SessionManagement00.log
User Session Limit	PASSED	SessionManagement01.log
Malform 01	PASSED	XML_testing.log
Malform 02	PASSED	XML_testing.log
Malform 03	PASSED	XML_testing.log
Malform 04	PASSED	XML_testing.log
Malform 05	PASSED	XML_testing.log
Malform 06	PASSED	XML_testing.log
Malform 07	PASSED	XML_testing.log
Verify_OAT_doesnot_invoke_password	PASSED	VerifyOATPassword.log
Verify_manual_backend_donot_invoke_password	PASSED	VerifymanualPassword00.log
Initiate task daemon via OAT	PASSED	javaTaskDameon01.log
Verify_Workbook_Limit_Per_Group_Template.txtRSC	FAILED	Verify_Workbook_Limit_Per_Group_Template.txt.err.log
Verify_Workbook_Limit_Per_Template.txtRSC	FAILED	Verify_Workbook_Limit_Per_Template.txt.err.log
Verify_Workbook_Limit_Per_User_Template.txtRSC	FAILED	Verify_Workbook_Limit_Per_User_Template.txt.err.log
Verify_Workbook_Template_Measure_Rights.txtRSC	FAILED	Verify_Workbook_Template_Measure_Rights.txt.err.log
Verify_Workbook_Template_Rights.txtRSC	FAILED	Verify_Workbook_Template_Rights.txt.err.log
Setup for fuzzing test for edge	PASSED	FuzzingTestSetupEDGE.log
Verify_Workbook_Limit_Per_Group_Template.txtRSC	FAILED	Verify_Workbook_Limit_Per_Group_Template.txt.err.log
Verify_Workbook_Limit_Per_Template.txtRSC	PASSED	
Verify_Workbook_Limit_Per_User_Template.txtRSC	PASSED	
Verify_Workbook_Template_Measure_Rights.txtRSC	PASSED	
Verify_Workbook_Template_Rights.txtRSC	PASSED	

Figure 10.3: HTML report file generated

Table 10.1: Overall hours estimation for server side automation

Reported On	2017
Total available tests	1269
Execution time with automation	70.9
Execution time without automation	793

- Test Result:- This contains the detail.html file
- Scripts:- This contain perl scripts for each test suite.
- Results:- Contains all the log files
- Lib:- Contains the Module.pm module and other library files
- Data:- This folder contains the output and input files.
- Domain:- Contains the domain files on which scripts are run
- Config and data files
- testDriver.pl:- Wrapper scripts to run all tests together
- attributes.ini:- It is the initialization file that contains all the variable and it's value which are going to be remain constant during execution.

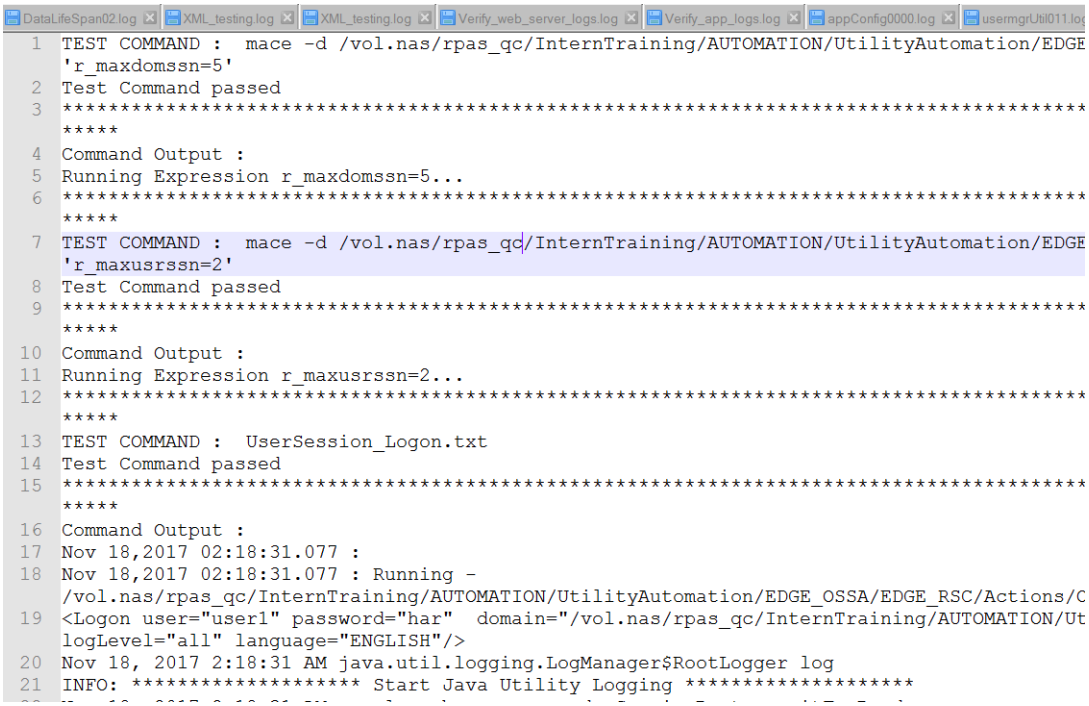


Figure 10.4: Generated log file

Input files contain the expected outcome of a particular test which is matched against the actual output which is recorded in the output folder.

After the tests are run, the output as shown in Figure 10.2 gives the status of a test i.e whether it has passed or failed and also the total number tests run, the number of tests failed and number of tests that passed.Also the execution time is shown in the output.

Also a HTML file is generated which is shown in Figure 10.3 has a list of tests, their status and link to their log files.These log files get a testname.log extension if test passed and testname.err.log extension when test gets failed.The log files looks as shown in figure 10.4.

Table 10.1 depicts the clear picture of the hours saved with the help of the test automation framework used.The time taken in manually testing the functionalities and time taken after automating them has a vast difference.Thus a lot of time is saved in automating these functionalities.

10.3 Client Side Reporting

In synergy, the reporting and logging is quite simpler as compared to server side reporting. In depth logs are generated on run time while executing the tests. The time taken to run the entire test suite, number of tests passed/failed, the time taken to execute each step in a test everything is listed in the run results. The figure 10.5 shows a sample run result to show what all information is listed while running the tests.

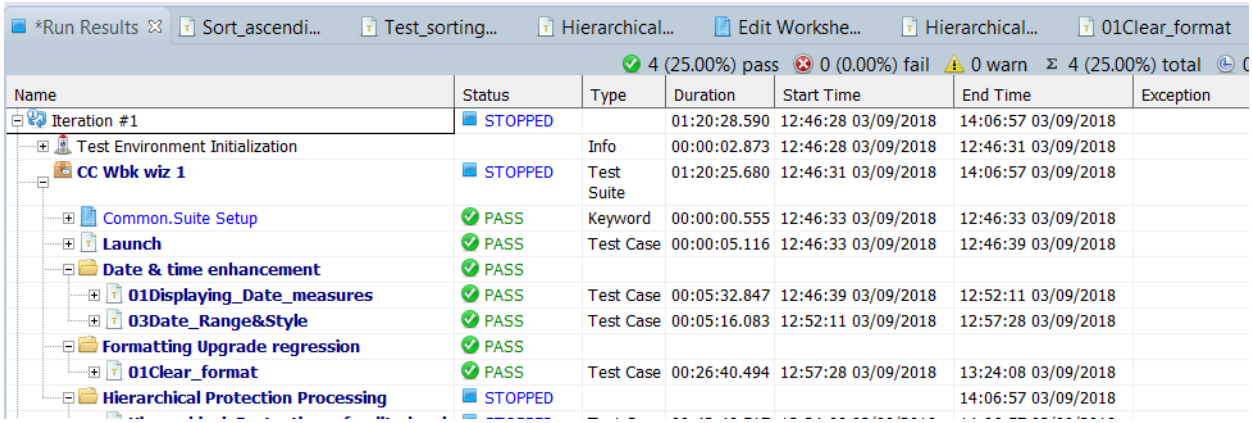


Figure 10.5: Run results

Table 10.2 depicts the clear picture of the hours saved with the help of the test automation framework used.The time taken in manually testing the functionalities and time taken after automating them has a vast difference.

Table 10.2: Overall hours estimation for client side automation

Reported On	2017
Total available tests	1748
Execution time with automation	79.9
Execution time without automation	694

Chapter 11

Conclusion

Software testing is evaluating a software application to test the deflection between a given input and an expected outcome. Through testing, the quality of the product is assured and also ensures that all the functionalities are behaving as expected or not.

Manual software testing is done by carefully analyzing and going through application screens in front of computers for various input combinations. Also the outcomes are compared with the expected behavior and recording their observations.

By automating we have good reporting & debugging options that allow the user to pinpoint and fix failures quickly.Also we can ensure that all the security functionalities are working in an expected way or not.By automating the test cases, they can be executed repeatedly with little or no manual effort. This saves a lot of time and also these scripts can be reused again and again for different versions of the software.The portability feature of Perl also adds up to the advantage of automating the tests.

Thus test automation is the most efficient way to enhance the effectiveness and coverage of the software testing.Also we can ensure that all the application's functionality are working in an expected manner.

Bibliography

[1] Jesús Chóliz, Julián Vilas, and José Moreira. Independent security testing on agile software development: a case study in a software company. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, pages 522–531. IEEE, 2015.

[2] Eliane Figueiredo Collins et al. Software test automation practices in agile development environment: An industry experience report. In *Proceedings of the 7th International Workshop on Automation of Software Test*, pages 57–63. IEEE Press, 2012.

[3] Barrett Gaines. Oracle® retail predictive application server administration guide for the classic client–volume 1, release 13.2. 2010.

[4] Karen M Goertzel, Theodore Winograd, Holly L McKinley, Lyndon J Oh, Michael Colon, Thomas McGibbon, Elaine Fedchak, and Robert Vienneau. Software security assurance: A state-of-art report (sar). Technical report, INFORMATION ASSURANCE TECHNOLOGY ANALYSIS CENTER (IATAC) HERNDON VA, 2007.

[5] Bernadette Goodman. Oracle® retail predictive application server administration guide, release 13.0. 2. 2008.

[6] Bernadette Goodman. Oracle® retail predictive application server administration guide for the classic client–volume 1, release 13.2. 2010.

[7] Sumit Jeloka, Lakshmi Kethana Kalyanasundaram, Andrew Koyfman, Nina Lewis Van Le, Stella Li, Janaki Narasinghanallur, Vikram Pesati, Andy Philips, Richard Smith, Deborah Steiner, et al. Oracle database advanced security administrator's guide 10g release 2 (10.2) b14268-02.

[8] E Michael Maximilien and Laurie Williams. Assessing test-driven development at ibm. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 564–569. IEEE, 2003.

[9] Sílvia Cristina dos Anjos Seabra Monteiro et al. Analysis of an oracle rpas planning solution. 2008.

[10] Sílvia Cristina dos Anjos Seabra Monteiro et al. Analysis of an oracle rpas planning solution. 2013.

[11] Frauke Paetsch, Armin Eberlein, and Frank Maurer. Requirements engineering and agile software development. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on*, pages 308–313. IEEE, 2003.

[12] Bruce Potter and Gary McGraw. Software security testing. *IEEE Security & Privacy*, 2(5):81–85, 2004.

[13] Hitesh Tahbildar and Bichitra Kalita. Automated software test data generation: direction of research. *International Journal of Computer Science and Engineering Survey*, 2(1):99–120, 2011.