Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK)

**Test plan**

**Version 1.0**

4/12/2020

**Document Control**

**Approval**

The Guidance Team and the customer shall approve this document.

**Document Change Control**

|  |  |
| --- | --- |
| Initial Release: | 4/10/2020 |
| Current Release: | 4/10/2020 |
| Indicator of Last Page in Document: | $ |
| Date of Last Review: | 4/16/2020 |
| Date of Next Review: | 4/21/2020 |
| Target Date for Next Update: | 4/21/2020 |

**Distribution List**

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:

Dr. Gates

Dr. Salamah

Dr. Roach

Elsa Tai Ramirez

Peter Hanson

Customer:

Dr. Oscar Perez

Vincent Fonseca

Herandy Denisse Vazquez

Baltazar Santaella

Florencia Larsen

Erick De Nava

Software Team Members:

Charlie Juarez

Miriam Juarez

Angelica Marquez

Andrew Munoz

Aaron Rodriguez

**Change Summary**

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 4/12/2020 | Angelica Marquez | Document creation |
| 1.1 | 4/14/2020 | Andrew Munoz, Aaron Rodriguez,  Angelica Marquez | 3. Testing approach tables |
| 1.2 | 4/15/2020 | Andrew Munoz | Section 1.1, section 1.4, section 1.6, section 6 (partial) |
| 1.3 | 4/16/2020 | Charlie Juarez | Section 4 test cases, Section 6 description |
| 1.4 | 4/16/2020 | Angelica Marquez | Section 1.5 |

Note: The template presented in this document was taken from:

Donaldson, S., and S. Siegel, *Successful Software Development*. Upper Saddle River, NJ: Prentice Hall, 2001, pp. 321-323.

Note: The template presented in this document was taken from: Donaldson, S., and S. Siegel, *Successful Software Development*. Upper Saddle River, NJ: Prentice Hall, 2001, pp. 321-323 and modified by Humberto Mendoza and Steve Roach.

Supplementary information is from:

Pfleeger, S. *Software Engineering, Theory and Practice*. Upper Saddle River, NJ: Prentice Hall, 1998, p. 365.

**Table of Contents**

[**DOCUMENT CONTROL II**](#_heading=h.30j0zll)

[**Approval ii**](#_heading=h.2xcytpi)

[**Document Change Control ii**](#_heading=h.1ci93xb)

[**Distribution List ii**](#_heading=h.3whwml4)

[**Change Summary ii**](#_heading=h.2bn6wsx)

[**1.**](#_heading=h.3dy6vkm) **INTRODUCTION 1**

[**1.1.**](#_heading=h.1t3h5sf) **Purpose 1**

[**1.2.**](#_heading=h.4d34og8) **Scope 1**

[**1.3.**](#_heading=h.2s8eyo1) **System Overview 1**

[**1.4.**](#_heading=h.17dp8vu) **Suspension and Exit Criteria 1**

[**1.5.**](#_heading=h.3rdcrjn) **Document Overview 1**

[**1.6.**](#_heading=h.26in1rg) **References 1**

[**2.**](#_heading=h.qsh70q) **TEST ITEMS AND FEATURES 2**

[**3.**](#_heading=h.35nkun2) **TESTING APPROACH 3**

[**4.**](#_heading=h.44sinio) **TEST XX 4**

[**4.1.**](#_heading=h.2jxsxqh) **Test <<test id>> 4**

[**5.**](#_heading=h.3as4poj) **USER INTERFACE TESTING 5**

[**6.**](#_heading=h.3j2qqm3) **TEST SCHEDULE 6**

[**7.**](#_heading=h.1y810tw) **OTHER SECTIONS 7**

[**8.**](#_heading=h.1pxezwc) **APPENDIX 8**

# **Introduction**

## **Purpose**

The purpose of creating the test plan document is to report the testing approach that will be used on the project. It will facilitate the task of testing and help organize the development and test effort to assess the system. The document contains the information on testing items and features, testing approach, test cases, and the test schedule to accomplish this purpose. The intended audience are the software engineers that will be implementing the test plan and the analysts that will be using the software, for future verification and maintenance.

## **Scope**

The PMR Insight Collective Knowledge (PICK) System will facilitate the current node correlation process for analysts to complete their cyber security attack graphs. Currently, the analysis process takes months to complete and the system is needed to shorten the completion time to a more reasonable length. The system will save the analysts time with the automatization of log file processing along with filtering/searching functionality. The system’s attack graph creator component will further save analysts time and provide a flexible structure for analysts to work off of. The purpose of PICK is to retrieve and layout log information provided by the adversarial assessment. The software will ingest the log files to the analyst's configuration and allow the analyst to produce attack graphs containing the events that transpired to provide a report for LSH.

## **System Overview**

<<Describe the system to be exercised by the testing approach specified in the plan. This overview serves to identify aspects of the system operation that will be the focus of the plan’s testing approach. This should align with the systems overview of other documents in the project.>>

## **Suspension and Exit Criteria**

The suspension criteria describe when we suspend testing, to be resumed at a later time. There is no suspension criteria in this test plan, meaning that all the test cases will be executed. The exit criteria indicate when testing stops. The exit criteria for this test plan is based on a high pass rate (number of test cases passed divided by number of test cases run). All critical tests must pass, and 90% of the non-critical tests must pass to stop testing.

## **Document Overview**

The Test Plan encompases Test Suites which describes test cases, and the importance of the test. Defines test objectives, test criteria, resource planning, and a detailed summary of all resources required of complet it, including human, equipment, and material. Also, the plan test environment with the schedule and estimation, for example project deadline and resources availability.

## **References**

[1] S. Roach et al, Software Requirements Specification, Lethality, Survivability, and HSI Directorate (LSH), 2019.

# **Test Items and Features**

<< This section describes the test items (e.g., components, classes, functions or methods) and the features to be tested. It may also list features not to be tested. A class diagram is useful. A table of features is useful. >>

# **Testing Approach**

<<Describe the approach to be used to the test the system. This description includes specifying the types of tests to be performed, e.g., tests designed to exercise system functions one by one; tests designed to exercise sequences of functions that approximate operational use of the system; tests designed to stress the system to its design and requirements limits. The description lists the specific tests to be performed, but does not give the test steps. For each of these tests, give it a name and specify its objective. Label the criticality of the test cases. >>

**Table 1: Ingestion Test Plan**

|  |  |  |
| --- | --- | --- |
| **TEST SUITE Ingestion** | | |
| **Description of Test Suite** | The following ingestion test plan will be conducted in order to ensure the application does not result in failures in regards to the ingestion process. | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| TSI OC1 | Status of a log file to “pass” in response to data ingestion operation is complete. | Critical |
| TSI OC2 | Do not ingest a file in response to a failed validation. | Critical |
| TSI OC3 | Ingest a file in response to a passed validation. | Critical |
| TSI OC4 | Allow a non-validated file to become ingestible if the user changes the non-validated file to become validated. | Critical |

**Table 2: Validation Test Plan**

|  |  |  |
| --- | --- | --- |
| **TEST SUITE Validation** | | |
| **Description of Test Suite** | Data validation test plan will be performed to ensure correctness of inspecting data in the clansed log files based on predefined data validation rules. | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| TSV OC1 | The cleansing script is applied to the log files. | Critical |
| TSV OC2 | Validate a file if it passes the cleansing script. | Critical |
| TSV OC3 | Only validate files that the user pointed to in the configuration. | Critical |
| TSV OC4 | Do not validate a file if the cleansing script fails. | Critical |

**Table 3: Filter Control Test Plan**

|  |  |  |
| --- | --- | --- |
| **TEST SUITE Filter Control** | | |
| **Description of Test Suite** | The filter control test plan will be performed to ensure the provided filtering components in the application are functional and correct. | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| TSFC OC1 | Filter log entries in response to the appropriate filter inputs from the user. | Normal |
| TSFC OC2 | Provide functional regex keyword search filter option. | Critical |
| TSFC OC3 | Do not filter any log entries if the user has no filter inputs activated. | Normal |
| TSFC OC4 | Ensure all filter inputs are accessible in every UI window they must exist in. | Critical |

**Table 4: Database Management Test Plan**

|  |  |  |
| --- | --- | --- |
| **TEST SUITE Database Management** | | |
| **Description of Test Suite** | The database management test plan will be conducted to guarantee the database is accurately managing the provided data. | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| TSDM OC1 | Create the database in response to the user using the application for the first time. | Critical |
| TSDM OC2 | Update the appropriate db table in response to the user editing an item from that db table in the system. | Critical |
| TSDM OC3 | Display UI tables with data from the appropriate db table data. | Critical |
| TSDM OC4 | Output appropriate requested data from the db table. | Critical |

**Table 5: Graph Control Test Plan**

|  |  |  |
| --- | --- | --- |
| **TEST SUITE Graph Control** | | |
| **Description of Test Suite** | The Graph Control test plan will be performed to ensure the correct content display, and exportation of the graph | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| TSGC OC1 | Graph accurately displays the specified nodes and relationships . | Critical |
| TSGC OC2 | Graph is exported to the format specified by the user. | Normal |
| TSGC OC3 | Graph displays content associated with the specified vector. | Critical |
| TSGC OC4 | Graph accurately displays the chosen visible attributes of a node. | Normal |

# **Test XX**

## Test TSI OC 3

**Objective: Ingest Log Entries**

**Notes:** This test assumes nothing went wrong in the validation process.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 1 | | | | Current Status: Pending | | |
| Test title: Ingest the log entries specified in the ingestable list. | | | | | | |
| Testing approach: The tester must have previously validated the log files which generate the log entries. | | | | | | |
| STEP  1 | OPERATOR ACTION  The tester presses the “Ingest Log Entries and Continue to Navigator” button. | PURPOSE  Determine if the program is able to ingest the log entries.. | | | EXPECTED RESULTS  The program should take the log entries and ingest them into MongoDB. In which the program will move on to the navigator window. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## **Test** TSV OC 1

**Objective: Cleanse Log Files**

**Notes:** This test assumes nothing went wrong in the configuration window and has provided all inputs necessary before moving on to the validation and ingestion window.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 1 | | | | Current Status: Pending | | |
| Test title: Cleanse the log files specified in the cleanse table. | | | | | | |
| Testing approach: The tester must have previously selected a non empty red, blue, and white team folder from which the program can extract the log files to cleanse. | | | | | | |
| STEP  1 | OPERATOR ACTION  The tester presses the cleanse button first in the window. | PURPOSE  Determine if the program is able to cleanse the log files on the cleanse table. | | | EXPECTED RESULTS  In the cleanse table the cleansed value should have changed from no to yes and in the validate table the cleansed values should have appeared on the validate table with the validated value as no. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test TSV OC 2

**Objective: Validate Log Files**

**Notes:** This test assumes nothing went wrong in the cleansing process after pressing the cleanse button.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 1 | | | | Current Status: Pending | | |
| Test title: Validate the log files specified in the validate table. | | | | | | |
| Testing approach: The tester must have previously pressed the cleanse button and have the cleansed files displayed on the validate table. | | | | | | |
| STEP  1 | OPERATOR ACTION  The tester presses the validate button. | PURPOSE  Determine if the program is able to validate the log files on the validate table. | | | EXPECTED RESULTS  The system should prompt the user for their Splunk credentials. | COMMENTS |
| 2 | The tester correctly inputs their Splunk credentials. | Determine if the Splunk upload is possible. | | | The system should have uploaded the log files to splunk and should have returned the log entries derived from the log files. The log entries will then be displayed on the ingestable list. In the validate table the validated value should have changed from no to yes. |  |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test TSFC OC 2

**Objective: Functional Regex**

**Notes:** This test assumes there is a log entry with the keyword inside of it.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 1 | | | | Current Status: Pending | | |
| Test title: Test the regex keyword search filter option. | | | | | | |
| Testing approach: The tester must be on the navigator window of an active project. | | | | | | |
| STEP  1 | OPERATOR ACTION  The tester provides a word that is guaranteed to be in a log entry in the regex bar. | PURPOSE  Determine if the program is able to search and display all log entries with the keyword provided. | | | EXPECTED RESULTS  In the navigation table the log entries with the searched keyword are displayed. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

# **User Interface Testing**

<<This section focuses on the interaction between the user and the system. For testing the user interface, consider the following traits:

* Consistent terminology, shortcut keys, menu selections, and presentation
* Correct language, spelling, and grammar.
* Flexibility in navigation between windows and interface elements.
* Error handling that will inform user of critical operations.
* Follows standards and guidelines such as placement of scroll bars, windows, and menu items.

This section could be integrated into Section 4.

>>

# **Test Schedule**

The following table contains the testing order in which the tests will be completed. The table contains the task and date on the left column for when and what to be completed. Along with the respective people who will be responsible for completing the task, appended by a description of the test responsibility.

|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| TSGC OC1-2 5/9/20 | Andrew Munoz, Charlie Juarez | Test TSG OC1 and OC2, the graph displaying nodes and relationships, and the graph export format. |
| TSG OC3-4 5/9/20 | Aaron Rodriguez, Angelica Marquez | Test TSG OC3 and OC4, the graph displaying specified vector, and the graph visible attributes. |

# **Other Sections**

<< Other sections that may appear in a test plan (but not required for this course) are:

* Test Management Requirements: how testing is to be managed; a delineation of responsibilities of each project organization involved with testing
* Staffing and training needs: delineate the responsibilities of those individuals who are to perform the testing, level of skill required, and training to be provided
* Environmental Requirements: describe the hardware (including communication and network equipment) needed to support testing; describe configuration of hardware components on which software and database to be tested are to operate.
* Software Requirements: describe the software needed to support testing; include the software code and databases that are object of the testing. Also include software tools such as compilers, CASE instruments and simulators that are needed to model the user’s operational environment.
* Risk and contingencies
* Cost: include an estimate of costs.
* Approvals
* Test Deliverables

>>

# **Appendix**

<< possibly more readable to put the expected output here and refer to it in the previous sections. Might also provide explicit directions for analysis of output, if it’s easier to read as an appendix or if analysis is post execution. >>

$$$$$$