Keikaku 企画

PMR Insight Collective Knowledge (PICK)
Test Plan

Version 0.2
4/16/2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

Initial Release:	0.1
Current Release:	0.2
Indicator of Last Page in Document:	[END]
Date of Last Review:	4/15/2020
Date of Next Review:	4/16/2020
Target Date for Next Update:	4/16/2020

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. Ann Gates, Dr. Steve Roach, Mr. Jake Lasley

Clients:
Mr. Vincent Fonseca, Mr. Baltazar Santaella, Ms. Herandy Vazquez, and Mr. Erick De Nava

Software Team Members:
Mr. Anthony Desarmier, Mr. Angel Villalpando, Mr. Mario Delgado, Mr. David Rayner,
Mr. Valentin Becerra, Mr. Jorge Garcia

Change Summary

The following table details changes made between versions of this document

Version	Date	Modifier	Description
0.1	04/07/2020	Anthony DesArmier	Added Template
0.2	04/14/2020	Angel Villalpando	Completed sections 1.1-1.4
0.2	04/15/2020	David Rayner	Completed 1.5, added suite to section 3, and two test cases to section 4.
0.2	4/15/2020	Anthony DesArmier	Formatting, grammar

TABLE OF CONTENTS

1. Introduction

1.1. Purpose

The purpose of this document is to outline the Test Plan for the PMR Insight Collective (PICK) system. This document will include the organizational responsibilities, the test approach, and the test schedule. This document will primarily discuss testing from the customer's point of view and should not be considered a general testing strategy, an integration test plan, or a unit test plan. By conducting the test cases proposed in this document, the customer should be able to demonstrate that the system performs that which it is intended to do.

1.2. Scope

The PMR Insight Collective Knowledge (PICK) is the software system for which this Test Plan is written for. PICK is a software system to help Prevent, Mitigate, and Recover Analysts analyze vast amounts of data collected during an Adversarial Assessment (AA) by allowing them to quickly search through, view, correlate, and build visual documents which help explain the AA itself to uninvolved personnel. The customers - in this case PMR Analysts - currently must sift through the vast amounts of generated data from the AA by hand which severely hinders their workflow and efficiency in developing a report with visual aids for which to explain the nature of the AA to other personnel.

PICK will allow the customers to insert all the data generated from an AA into its system and display an organized, searchable database of that information. The customers can then quickly and efficiently find and correlate relevant data events together and help craft timelines which describe the significant events and their relations to one another during the AA. PICK will then assist the customers in crafting a visual representation of these series of events as attack graphs in order to help visualize the timeline of the AA. This assistance of analyzing the data generated by the AA and constructing visual representations of significant events will substantially reduce the time and work hours needed by the customers to understand and construct a report on the results of the AA to deliver to other personnel.

1.3. System Overview

The PICK system utilizes several python libraries for the graphical user interface which must be tested to ensure that they perform their desired tasks. Additionally, the system heavily interacts with the Splunk Extract, Load, and Transform (ELT) system. The interaction with this system requires testing to ensure that the data sent to a from it follow the specifications outlined by the design. Finally, ensuring that the system correctly creates vectors, each with respective log entries, is important to the overall success of this system. These items are the focal points for the testing outlined in this document.

1.4. Suspension and Exit Criteria

If at any point a critical test fails, testing will be suspended. Critical tests are intended to assess the functionality of the major components within the system. If any of these major components are not functioning as intended, several subsequent tests dependent on this component will also fail or will not be testable. For this reason, testing shall be suspended source code redeveloped to restore functionality to such major components.

Once all critical tests have passed, testing shall be complete. Once critical components are exhibiting the desired behaviors, then the system satisfies the core requirements laid out in the initial specification of the system.

1.5. Document Overview

The test plan document consists of the following sections:

Introduction:
This section describes the overview of the testing plan. It includes the purpose of the document, the overall scope of the project, and the suspension, exit criteria regarding system tests to be run.

Test Items and Features:
This section describes the testing items (e.g. components, classes, functions or methods) and the features to be tested.

Testing Approach:
This section describes the testing approach we the development team are to establish. The type of tests to be run in order to test system functions. Each test is to contain a description and unique test identifier.

Test cases:
This section describes the tests that were run, including test input, test procedures and outcomes. Each test is divided by the following sections: test number, current status, title, approach, step, operator action, purpose, expected results, comments, remarks, conclusion, date completed, and team that performed the test.

User Interface Testing:
This section describes the interaction between the system and user components Including consistent terminology, shortcut keys, menu selections, and presentation, flexibility in navigation between windows and interface elements and potential error handling that will inform user of critical operations.

Test Schedule:
This section describes the completion dates of each test.

Other:
This section describes the other potential test documentation such as:

- 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:
References of expected output and explicit directions for analysis of output.

1.6. References

<<List all the references applicable to the test plan. Generally, this includes project standards, SRS, SDD, and a product assurance plan.>>

2. Test Items and Features

Feature: File Ingestion
Class: Validator
Class: SplunkManager

Feature: File Cleansing
Class Validator

Feature: File Validation
Class: Validator
Class: EnforcementActionReport

Feature: Log entry to vector assignment
Class: LogEntry
Class: IDDict
Class: Vector

Feature: Sort/Filter log entries and nodes
Class: Sort
Class: Filter

Feature: Export vector table
Class: ExportGraph
Class: Vector

Feature: Export vector graph
Class: ExportTable
Class: Vector

Feature: Graphing
Class: GraphEditor
Class: GraphEditorScene
Class: GraphEditorView
Class: GraphEditorWindow
Class: NodeItem
Class: RelationshipItem
Class: VectorItemGroup

Feature: Search and Filter
Class: Sort
Class: Filter

Feature: Data storage

Feature: Lead-Host data management
Class: Sync
Class: ProjectMerge

Feature: Commit management
Class: History

3. Testing Approach

Table 1: Test Plan

TEST SUITE <Start Ingestion>		
Description of Test Suite	The following test suite is to evaluate the functionality of the start ingestion process the system is to perform.	
Test Case Identifier	**Objective**	**Criticality**
ING OC1	Open Event configuration dialog in response to File->Event selection.	Critical
ING OC2	Save Event configuration (name, description, start, and end times) in response to save button clicked.	Critical
ING OC3	Open directory configuration in response to Directory button clicked.	Critical
ING OC4	Start ingestion process once valid directories (root, red, white, and blue) specified.	Critical
ING OC4	Create copies of root directory files.	Critical
ING OC6	Initiate cleansing operation on root directory files.	Critical
ING OC7	Initiate validating operation on cleansed root directory files.	Critical
ING OC8	Initiate ingestion operation on validated root directory files.	Critical
ING OC9	Generate enforcement action reports for invalid (non-ingested) files.	Normal
ING OC10	Populate log file table with file statuses (cleansed, validated, ingested) and enforcement action reports if applicable.	Critical
ING OC10	Populate log entry table with ingested parsed entries.	Critical

4. Tests

4.1. Start Ingestion

Objective: To establish proper functionality of the start ingestion process.
Notes: Access to different test files with various

<table>
<tr><td colspan="3">Test No.: ING OC2</td><td colspan="2">Current Status: Passed</td></tr>
<tr><td colspan="5">Test title: Save event details (name, description, start time, end time)</td></tr>
<tr><td colspan="5">Testing approach: This test will be conducted on the event configuration dialog, field inputs are selected and then output messages are observed.</td></tr>
<tr><td>STEP</td><td>OPERATOR ACTION</td><td>PURPOSE</td><td>EXEPCTED RESULTS</td><td>COMMENTS</td></tr>
<tr><td>1</td><td>Begin test, click “save event” button with input fields name and description empty.</td><td>Initial Condition</td><td>Prompt stating “name or description” input fields empty.</td><td></td></tr>
<tr><td>2</td><td>Enter text in name field, but leave description field empty. Click “save event” button.</td><td>Check with one field (name) empty.</td><td>Prompt stating “name or description” input fields empty.</td><td></td></tr>
<tr><td>3</td><td>Enter text in description field, but leave name field empty. Click “save event” button</td><td>Check with one field (description) empty.</td><td>Prompt stating “name or description” input fields empty.</td><td></td></tr>
<tr><td>4</td><td>Leave default (both start time and end time fields are the same). Click “save event”.</td><td>Check to see if time is in valid ranges. (start before end)</td><td>Prompt stating “invalid end time”.</td><td></td></tr>
<tr><td>5</td><td>Set start time after end time.</td><td>Check to see if time is in valid ranges. (start before end)</td><td>Prompt stating “invalid end time”.</td><td></td></tr>
<tr><td>6</td><td>Set end time before start time.</td><td>Check to see if time is in valid ranges. (start before end)</td><td>Prompt stating “invalid end time”.</td><td></td></tr>
<tr><td>7</td><td>All valid fields entered.</td><td>Check to see if event with valid fields is saved.</td><td>Prompt stating “event saved”</td><td></td></tr>
<tr><td colspan="5">Concluding Remarks:
Tests provided the correct response prompts.</td></tr>
<tr><td colspan="3">Testing Team: Keiaku</td><td colspan="2">Date Completed: 04/15/2020</td></tr>
</table>

Test No.: ING OC6		Current Status: Pending		
Test title: Initiate cleansing action on files				
Testing approach: This test will be conducted on the cleansing operation, two input files are selected one .csv file and one .log file.				
STEP	OPERATOR ACTION	PURPOSE	EXEPCTED RESULTS	COMMENTS
1	Add empty lines and invalid binary characters to .log file.	Initial condition	N/A	
2	Click “start ingestion” under directory dialog.	Start ingestion operation.	N/A	
3	Analyze log file table	Check to see if cleansing status is true “green checkmark”	Log file details populated on log file table and cleansing status is set.	Displayed correct status.
4	Analyze file contents	Check to see if file has been cleansed.	Files updated and stripped of empty lines/rows, and invalid binary characters.	.log file was stripped of empty lines and non-ascii characters.
Concluding Remarks: Other file formats (.csv, .pdf, image formats, and media formats) need to be tested.				
Testing Team: Keiaku		Date Completed: 04/15/2020		

5. Test Schedule

<< Specify the schedule for testing activities. A table with the order and completion dates of the tests is useful. The table below might be useful.>>

Task and date	People	Description

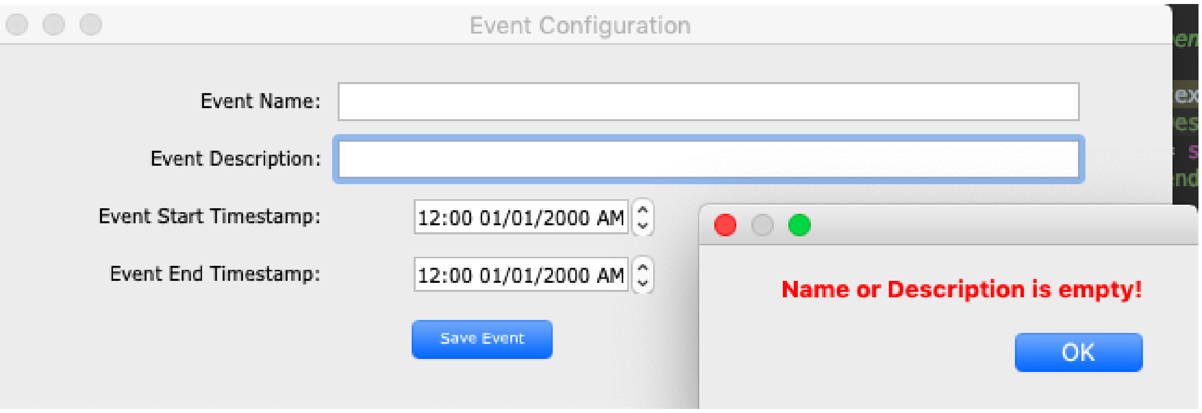
6. Other Sections

<< 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

>>

7. Appendix



Event Name:

Event Description:

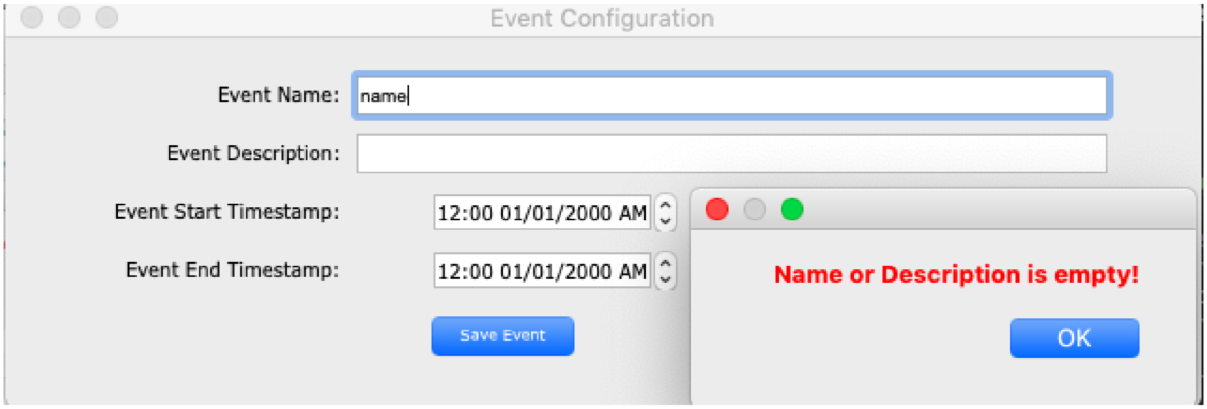
Event Start Timestamp: 12:00 01/01/2000 AM

Event End Timestamp: 12:00 01/01/2000 AM

Name or Description is empty!

Figure a. ING OC2

>>



Event Name: name

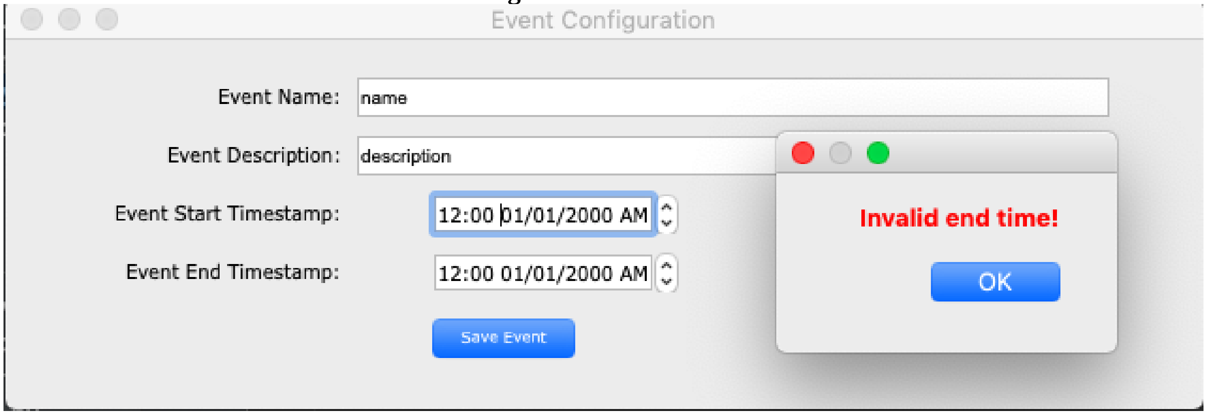
Event Description:

Event Start Timestamp: 12:00 01/01/2000 AM

Event End Timestamp: 12:00 01/01/2000 AM

Name or Description is empty!

Figure b. ING OC2



Event Name: name

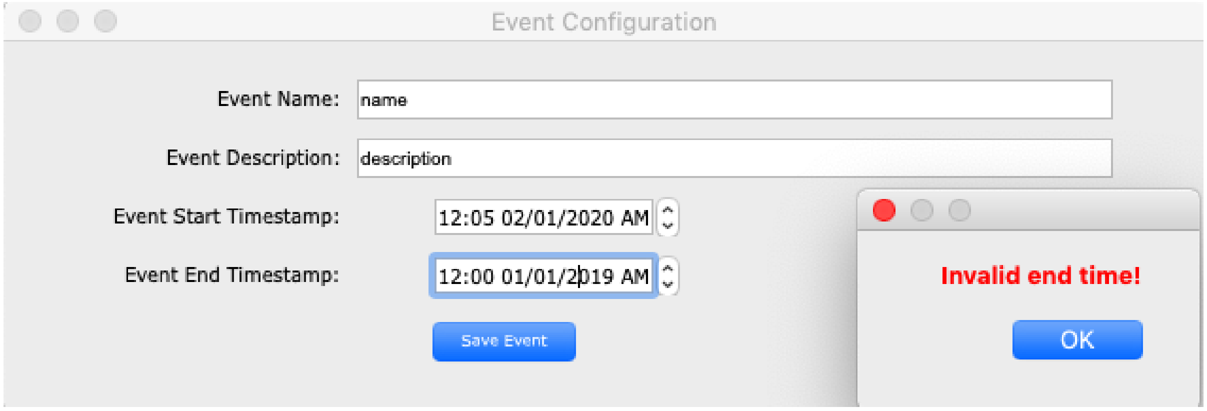
Event Description: description

Event Start Timestamp: 12:00 01/01/2000 AM

Event End Timestamp: 12:00 01/01/2000 AM

Invalid end time!

Figure c. ING OC2



Event Configuration

Event Name: name

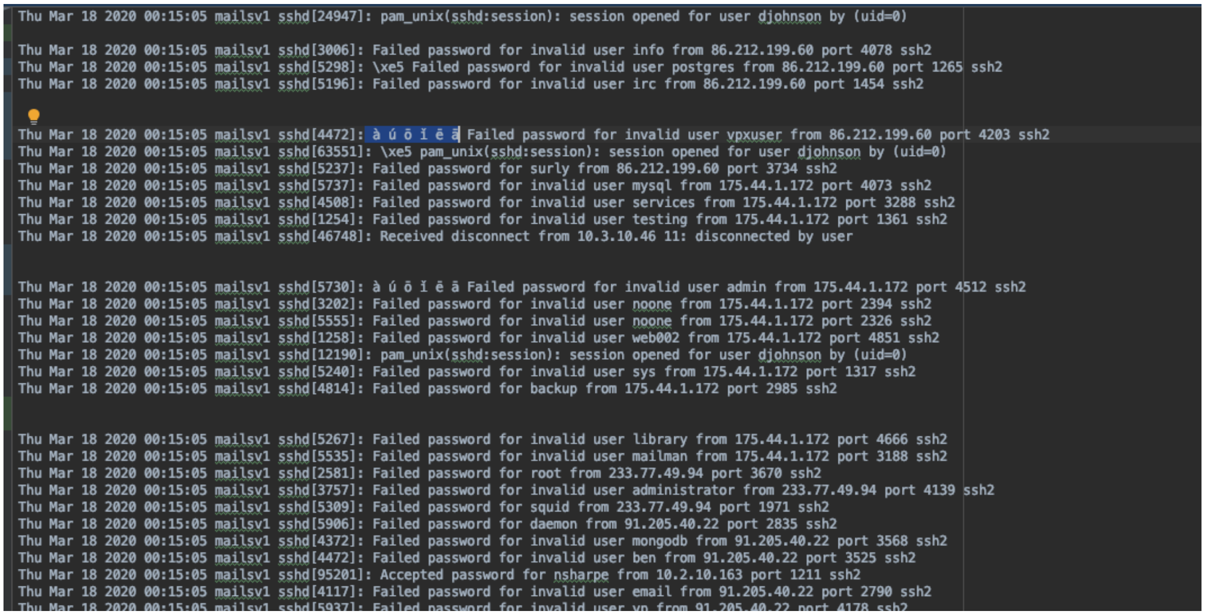
Event Description: description

Event Start Timestamp: 12:05 02/01/2020 AM

Event End Timestamp: 12:00 01/01/2019 AM

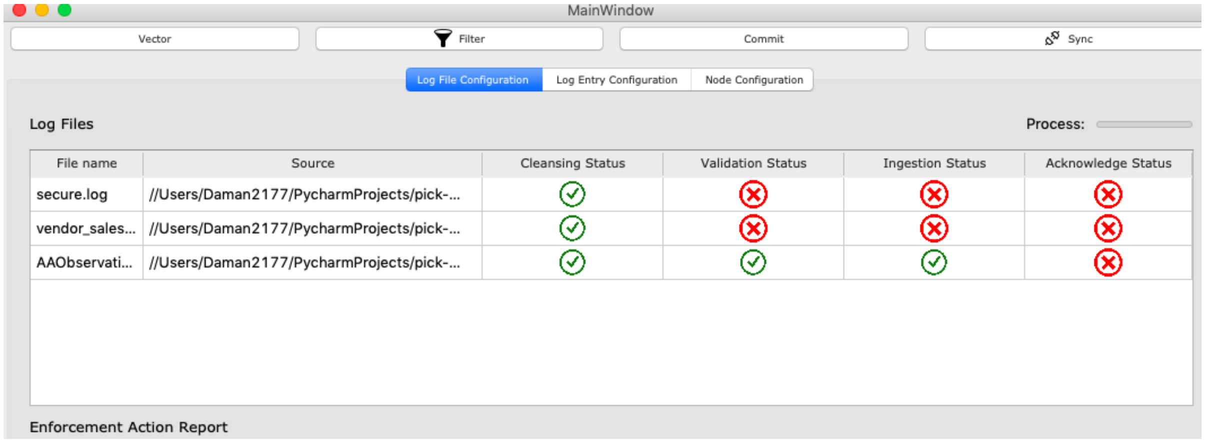
Invalid end time!

Figure d. ING OC2



```
Thu Mar 18 2020 00:15:05 mailsv1 sshd[24947]: pam_unix(sshd:session): session opened for user djohnson by (uid=0)

Thu Mar 18 2020 00:15:05 mailsv1 sshd[3006]: Failed password for invalid user info from 86.212.199.60 port 4078 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5298]: \xe5 Failed password for invalid user postgres from 86.212.199.60 port 1265 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5196]: Failed password for invalid user irc from 86.212.199.60 port 1454 ssh2


Thu Mar 18 2020 00:15:05 mailsv1 sshd[4472]: à ú ō ĭ ē ā Failed password for invalid user vpxuser from 86.212.199.60 port 4203 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[63551]: \xe5 pam_unix(sshd:session): session opened for user djohnson by (uid=0)
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5237]: Failed password for surly from 86.212.199.60 port 3734 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5737]: Failed password for invalid user mysql from 175.44.1.172 port 4073 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4508]: Failed password for invalid user services from 175.44.1.172 port 3288 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[1254]: Failed password for invalid user testing from 175.44.1.172 port 1361 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[46748]: Received disconnect from 10.3.10.46 11: disconnected by user


Thu Mar 18 2020 00:15:05 mailsv1 sshd[5730]: à ú ō ĭ ē ā Failed password for invalid user admin from 175.44.1.172 port 4512 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[3202]: Failed password for invalid user noone from 175.44.1.172 port 2394 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5555]: Failed password for invalid user noone from 175.44.1.172 port 2326 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[1258]: Failed password for invalid user web002 from 175.44.1.172 port 4851 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[12190]: pam_unix(sshd:session): session opened for user djohnson by (uid=0)
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5240]: Failed password for invalid user sys from 175.44.1.172 port 1317 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4814]: Failed password for backup from 175.44.1.172 port 2985 ssh2


Thu Mar 18 2020 00:15:05 mailsv1 sshd[5267]: Failed password for invalid user library from 175.44.1.172 port 4666 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5535]: Failed password for invalid user mailman from 175.44.1.172 port 3188 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[2581]: Failed password for root from 233.77.49.94 port 3670 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[3757]: Failed password for invalid user administrator from 233.77.49.94 port 4139 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5309]: Failed password for squid from 233.77.49.94 port 1971 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5906]: Failed password for daemon from 91.205.40.22 port 2835 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4372]: Failed password for invalid user mongodb from 91.205.40.22 port 3568 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4472]: Failed password for invalid user ben from 91.205.40.22 port 3525 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[95201]: Accepted password for nsharpe from 10.2.10.163 port 1211 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4117]: Failed password for invalid user email from 91.205.40.22 port 2790 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5937]: Failed password for invalid user vn from 91.205.40.22 port 4178 ssh2
```

Figure e. OC6 secure.log



MainWindow

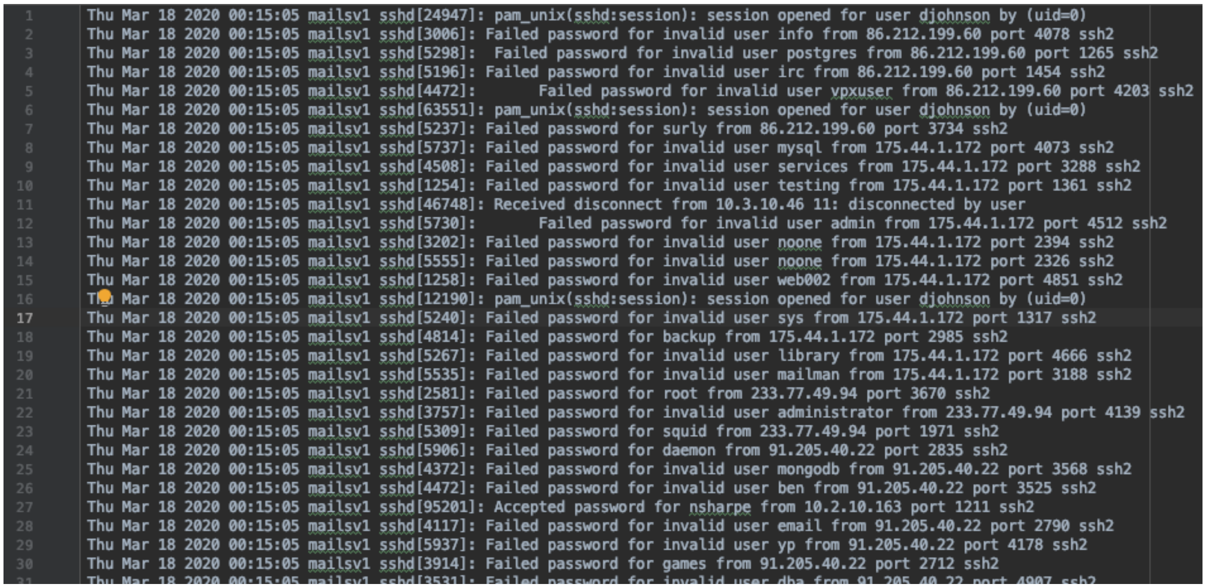
Log Files

Process:

File name	Source	Cleansing Status	Validation Status	Ingestion Status	Acknowledge Status
secure.log	//Users/Daman2177/PycharmProjects/pick-...	✓	☒	☒	☒
vendor_sales...	//Users/Daman2177/PycharmProjects/pick-...	✓	☒	☒	☒
AAObservati...	//Users/Daman2177/PycharmProjects/pick-...	✓	✓	✓	☒

Enforcement Action Report

Figure f. OC6



```
Thu Mar 18 2020 00:15:05 mailsv1 sshd[24947]: pam_unix(sshd:session): session opened for user djohnson by (uid=0)
Thu Mar 18 2020 00:15:05 mailsv1 sshd[3006]: Failed password for invalid user info from 86.212.199.60 port 4078 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5298]:  Failed password for invalid user postgres from 86.212.199.60 port 1265 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5196]: Failed password for invalid user irc from 86.212.199.60 port 1454 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4472]:       Failed password for invalid user vpxuser from 86.212.199.60 port 4203 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[63551]: pam_unix(sshd:session): session opened for user djohnson by (uid=0)
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5237]: Failed password for surly from 86.212.199.60 port 3734 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5737]: Failed password for invalid user mysql from 175.44.1.172 port 4073 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4508]: Failed password for invalid user services from 175.44.1.172 port 3288 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[1254]: Failed password for invalid user testing from 175.44.1.172 port 1361 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[46748]: Received disconnect from 10.3.10.46 11: disconnected by user
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5730]:       Failed password for invalid user admin from 175.44.1.172 port 4512 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[3202]: Failed password for invalid user noone from 175.44.1.172 port 2394 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5555]: Failed password for invalid user noone from 175.44.1.172 port 2326 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[1258]: Failed password for invalid user web002 from 175.44.1.172 port 4851 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[12190]: pam_unix(sshd:session): session opened for user djohnson by (uid=0)
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5240]: Failed password for invalid user sys from 175.44.1.172 port 1317 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4814]: Failed password for backup from 175.44.1.172 port 2985 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5267]: Failed password for invalid user library from 175.44.1.172 port 4666 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5535]: Failed password for invalid user mailman from 175.44.1.172 port 3188 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[2581]: Failed password for root from 233.77.49.94 port 3670 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[3757]: Failed password for invalid user administrator from 233.77.49.94 port 4139 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5309]: Failed password for squid from 233.77.49.94 port 1971 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5906]: Failed password for daemon from 91.205.40.22 port 2835 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4372]: Failed password for invalid user mongodb from 91.205.40.22 port 3568 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4472]: Failed password for invalid user ben from 91.205.40.22 port 3525 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[95201]: Accepted password for nsharpe from 10.2.10.163 port 1211 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[4117]: Failed password for invalid user email from 91.205.40.22 port 2790 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[5937]: Failed password for invalid user yp from 91.205.40.22 port 4178 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[3914]: Failed password for games from 91.205.40.22 port 2712 ssh2
Thu Mar 18 2020 00:15:05 mailsv1 sshd[3531]: Failed password for invalid user dba from 91.205.40.22 port 4907 ssh2
```

Figure g. OC6 secure.log