Test Plan and Results

Overall Test Plan

Our approach to testing is two-fold. First we will test the system components individually, using simulated data, covering a variety of normal and abnormal conditions. We will create various log files, regular expressions, and action rules specifically for testing. These will test both normal and abnormal cases. For the second phase we will test the entire system in a production environment, using some of Projetech's client servers. This will further test the system functionality and provide stress testing, to make sure that it can handle the load of a production environment.

1. test case identifier (a number or unique name)
2. purpose of test
3. description of test
4. inputs used for test
5. expected outputs/results
6. normal/abnormal/boundary case indication
7. blackbox/whitebox test indication
8. functional/performance test indication
9. unit/integration test indication

Python-Game Communication Test

1. Python server can send and receive data from the game
2. Transmitted data from the game represents the current game state
3. Transmitted data from the server represents the decisions of the agents in the form of key presses
4. Test that the server will not hang

Neural Network Test

1. Given an image of the game, the network can decide to press keys
2. The agents can make decisions that accomplish their goals

Latency Test

The time it takes to send data to the server and receive a reply must be sufficiently low

Unity-

1. Player and chaser collide with walls smoothly.
2. Player and chaser have a variable height jump.
3. When the player presses the "A", "D" keys, the player should move left and right.
4. When the player presses the "W", the player should jump.
5. When a chaser meets the evader, the game will reset.
6. A timer will display in the upper corner of the game screen to represent how long the evader has stayed alive (i.e. avoided the chaser(s))

7. A selection will be made by the user to decide which role they want to play in and what role they want the neural network to have.

Test Case Descriptions

CS1.1 **Central Server Test 1**

CS1.2 This test will ensure that the central server's query is correct.

CS1.3 This test will run the same query that the central server uses to pull records that need to be handled. This query will then be run against the database in SQL Management Studio to see the results.

CS1.4 Inputs: The inputs for this test will be the query used by the central server and unprocessed records in the database.

CS1.5 Outputs: All of the unprocessed records in the database will return correctly and none of the processed records will be returned.

CS1.6 Normal

CS1.7 Whitebox

CS1.8 Functional

CS1.9 Unit Test

CS1.10 Results: All unprocessed records were returned, no other records were returned.

CS2.1 **Central Server Test 2**

CS2.2 This test will test the processing of the various message types.

CS2.3 For this test we will run the central server and feed it entries through the database. This will simulate what the agent would normally do. We will give it entries of varying sizes and severities. This will test that the server can handle all the inputs that the agents will give it.

CS2.4 Inputs: Entries into the database that are just like the ones that the agent will write.

CS2.5 Outputs: Entries should be marked as processed and if the entry is severity of 1 or 2 the appropriate emails should be sent out.

CS2.6 Normal

CS2.7 Blackbox

CS2.8 Functional

CS2.9 Unit Test

CS2.10 Results: Entries were processed correctly and appropriate emails were sent.

CS3.1 **Central Server Test 3**

CS3.2 This test will test the deletion of old entries in the central database.

CS3.3 For this test we will test to seed it the central server can properly detect and delete entries in the database that have been processed and have been in the database for more than 30 days.

CS3.4 Inputs: Entries into the database that are marked as processed and that have timestamps that are more than 30 days old.

CS3.5 Outputs: All of the processed and old entries in the system should be deleted.

CS3.6 Normal

CS3.7 Blackbox

CS3.8 Functional

CS3.9 Unit Test

CS3.10 Results: All entries over 30 days old were deleted. All entries under 30 days old were maintained.

WA1.1 **Web Admin Test 1**

WA1.2 This test will test the ability for a user to authenticate against the LDAP servers.

WA1.3 We will test this by navigating to the web admin's various pages and ensure that it always redirects us to the login page. Once on the login page we will enter valid credentials and ensure that the user is logged in. Once logged in they should be able to browse the various pages freely.

WA1.4 Inputs: Valid credentials to log into the system

WA1.5 Outputs: A successful login and the ability to browse the pages freely.

WA1.6 Normal

WA1.7 Blackbox

WA1.8 Functional

WA1.9 Unit Test

WA1.10 Results: All pages properly redirect to login page. Login successfully authenticates to the LDAP server.

WA2.1 **Web Admin Test 2**

WA2.2 This test will test the ability for a user to authenticate against the LDAP servers.

WA2.3 We will test this by navigating to the web admin's various pages and ensure that it always redirects us to the login page. Once on the login page we will enter invalid credentials and ensure that the user is not logged in. The user should not be able to browse the various pages freely.

WA2.4 Inputs: Invalid credentials to log into the system

WA2.5 Outputs: A unsuccessful login and not ability to browse the pages freely.

WA2.6 Normal

WA2.7 Blackbox

WA2.8 Functional

WA2.9 Unit Test

WA2.10 Results: An error message was received and the user was not able to access any pages.

WA3.1 **Web Admin Test 3**

WA3.2 This test will test the home page where entries are viewed.

WA3.3 For this test we will navigate to the home page while logged in and ensure that all the entries that are in the database are visible.

WA3.4 Inputs: Logged in user

WA3.5 Outputs: Properly displayed entries from the database

WA3.6 Normal

WA3.7 Blackbox

WA3.8 Functional

WA3.9 Unit Test

WA3.10 Results: Database entries are correctly displayed.

WA4.1 **Web Admin Test 4**

WA4.2 This test will test the home page where entries are properly filtered.

WA4.3 For this test we will navigate to the home page while logged in and ensure that all the entries that are in the database are visible. Then apply various filtering criteria and ensure that the displayed results all meet the criteria. We will test all the different filtering options including date, severity and servers.

WA4.4 Inputs: Logged in user and filter criteria

WA4.5 Outputs: Properly displayed entries from the database that meet the selected filtering criteria

WA4.6 Normal

WA4.7 Blackbox

WA4.8 Functional

WA4.9 Unit Test

WA4.10 Results: Entries were correctly filtered by various criteria.

WA5.1 **Web Admin Test 5**

WA5.2 This test will ensure that when navigating to the log locations page that all of the entries in the database display properly.

WA5.3 A logged in user will navigate to the log locations page and ensure that all the entries that are displayed match the list of entries that are currently in the database. No entries should be missing or added.

WA5.4 Inputs: Logged in user

WA5.5 Outputs: Display all the log location entries in the database

WA5.6 Normal

WA5.7 Blackbox

WA5.8 Functional

WA5.9 Unit Test

WA5.10 Results: Log locations are displayed exactly as they are in the database.

WA6.1 **Web Admin Test 6**

WA6.2 This test will test that users can enter a new log location.

WA6.3 The user will navigate to the “log location” page, press the “new” button, enter the server name, log path, and log type, then press submit. This information will be entered into the database.

WA6.4 Inputs: Server name, log path, log type.

WA6.5 Outputs: Database contains server name, log path, log type.

WA6.6 Normal

WA6.7 Blackbox

WA6.8 Functional

WA6.9 Unit Test

WA6.10 Results: Information is correctly entered into the database.

WA7.1 **Web Admin Test 7**

WA7.2 This test will test that users can edit log location.

WA7.3 The user will navigate to the “log location” page, press the “edit” button corresponding to the entry they wish to edit, enter and edit the server name, log path, and log type, then press submit. This information will be entered into the database.

WA7.4 Inputs: Server name, log path, log type.

WA7.5 Outputs: Database contains edited server name, log path, log type.

WA7.6 Normal

WA7.7 Blackbox

WA7.8 Functional

WA7.9 Unit Test

WA7.10 Results: Information is correctly changed in the database.

WA8.1 **Web Admin Test 8**

WA8.2 This test will ensure that when navigating to the action rules page that all of the entries in the database display properly.

WA8.3 A logged in user will navigate to the action rules page and ensure that all the entries that are displayed match the list of entries that are currently in the database. No entries should be missing or added.

WA8.4 Inputs: Logged in user

WA8.5 Outputs: Display all the action rules entries in the database

WA8.6 Normal

WA8.7 Blackbox

WA8.8 Functional

WA8.9 Unit Test

WA8.10 Results: Action rules are displayed exactly as they are in the database.

WA9.1 **Web Admin Test 9**

WA9.2 This test will test that users can enter a new message type.

WA9.3 The user will navigate to the “message type” page, press the “new” button, enter the log type, severity, and regular expression, then press submit. This information will be entered into the database.

WA9.4 Inputs: Log type, severity, and regular expression.

WA9.5 Outputs: Database contains log type, severity, and regular expression.

WA9.6 Normal

WA9.7 Blackbox

WA9.8 Functional

WA9.9 Unit Test

WA9.10 Results: Information is correctly entered into the database.

WA10.1 **Web Admin Test 10**

WA10.2 This test will test that users can edit message type.

WA10.3 The user will navigate to the “message type” page, press the “edit” button corresponding to the entry they wish to edit, enter or change the log type, severity, and regular expression, then press submit. This information will be entered into the database.

WA10.4 Inputs: Log type, severity, and regular expression.

WA10.5 Outputs: Database contains edited log type, severity, and regular expression.

WA10.6 Normal

WA10.7 Blackbox

WA10.8 Functional

WA10.9 Unit Test

WA10.10 Results: Information is correctly changed in the database.

WA11.1 **Web Admin Test 11**

WA11.2 This test will ensure that when navigating to the action rules page that all of the entries in the database display properly.

WA11.3 A logged in user will navigate to the action rules page and ensure that all the entries that are displayed match the list of entries that are currently in the database. No entries should be missing or added.

WA11.4 Inputs: Logged in user

WA11.5 Outputs: Display all the action rules entries in the database

WA11.6 Normal

WA11.7 Blackbox

WA11.8 Functional

WA11.9 Unit Test

WA11.10 Results: Action rules are displayed exactly as they are in the database.

WA12.1 **Web Admin Test 12**

WA12.2 This test will test that users can enter a new action rule.

WA12.3 The user will navigate to the “action rule” page, press the “new” button, enter the message type, log type, and action, then press submit. This information will be entered into the database.

WA12.4 Inputs: Message type, log type, and action.

WA12.5 Outputs: Database contains message type, log type, and action.

WA12.6 Normal

WA12.7 Blackbox

WA12.8 Functional

WA12.9 Unit Test

WA12.10 Results: Information is correctly entered into the database.

WA13.1 **Web Admin Test 13**

WA13.2 This test will test that users can edit an action rule.

WA13.3 The user will navigate to the “action rule” page, press the “edit” button corresponding to the entry they wish to edit, edit the message type, log type, and action, then press submit. This information will be changed in the database.

WA13.4 Inputs: Message type, log type, and action.

WA13.5 Outputs: Database contains the new message type, log type, and action.

WA13.6 Normal

WA13.7 Blackbox

WA13.8 Functional

WA13.9 Unit Test

WA13.10 Results: Information is correctly changed in the database.

A1.1 **Agent Test 1**

A1.2 This test will ensure that the agent opens a log file and reads in its contents.

A1.3 We will add various correct and incorrect log locations to the database and check that either the log file is opened of the proper error is logged to the database.

A1.4 Inputs:

a) Correct path to a log that exists and can be opened.

b) Incorrect path to a log that does not exist.

c) Correct path to a log that exists but is protected.

A1.5 Outputs:

a) The agent reads in the contents of a file.

b) The agent logs an error to the database that the file does not exist.

c) The agent logs an error to the database that the file cannot be opened.

A1.6 Normal & Abnormal

A1.7 Blackbox

A1.8 Functional

A1.9 Unit

A1.10 Results: For the correct log path, the file contents were correctly processed. For the incorrect path, an error that the log does not exist was logged. For the write protected file, an error that the file cannot be opened was logged.

A2.1 **Agent Test 2**

A2.2 This test will ensure that the agent correctly parses the log entries.

A2.3 We will add various regular expressions to the message type table in the database. The agent will query the database to get the message types, apply the regular expressions to the log contents, and correctly parse the text.

A2.4 Inputs: regular expression and log text

a) Regular expression matches log text.
b) Regular expression does not match log text.

A2.5 Outputs:

a) Log entry is correctly parsed.
b) Error message is logged.

A2.6 Normal & Abnormal

A2.7 Blackbox

A2.8 Functional

A2.9 Unit

A2.10 Results: When the regular expression matches the text, the log is entered in the database, otherwise an error message is logged.

A3.1 **Agent Test 3**

A3.2 This test will ensure that the agent detects changes to the log file and enters the messages into the database.

A3.3 We will append entries to a log file that the agent has opened and check that they appear in the database.

A3.4 Inputs: Entries added to the log file.

A3.5 Outputs: Entries written to the database.

A3.6 Normal

A3.7 Blackbox

A3.8 Functional

A3.9 Unit

A3.10 Results: The entries were correctly written to the database.

FS1.1 **Full System Test 1**

FS1.2 This test will be used to see that the full system works together well

FS1.3 We will deploy the agent on an active server where it will be setup to monitor a log file. We then will compare the emails and entries that are produced to the actual log file that it is set to test on.

FS1.4 Inputs: Log file

FS1.5 Outputs: Appropriate emails and database entries based on the data in the log

FS1.6 Normal

FS1.7 Blackbox

FS1.8 Functional

FS1.9 Integration

FS1.10 Results: Log watcher successfully observed new log entries, parsed them, entered them in the database, and processed them.

FS2.1 **Full System Test 2**

FS2.2 This test will be used to see that the full system works together well

FS2.3 We will deploy the agent on multiple active server where it will be setup to monitor a different log files. We then will compare the emails and entries that are produced to the actual log files that it is set to test on. We will let this run for over a week to attempt to simulate the stressed that will actually be put on the system.

FS2.4 Inputs: Log files on different servers

FS2.5 Outputs: Appropriate emails and database entries based on the data in the logs

FS2.6 Normal

FS2.7 Blackbox

FS2.8 Performance

FS2.9 Integration

FS2.10 Results: Correct emails were sent and appropriate entries were written to the database.

Test Case Matrix

	Normal/ Abnormal	Blackbox/ Whitebox	Functional/ Performance	Unit/ Integration
CS1	Normal	White	Function	Unit
CS2	Normal	Black	Function	Unit
CS3	Normal	Black	Function	Unit
WA1	Normal	Black	Function	Unit
WA2	Normal	Black	Function	Unit
WA3	Normal	Black	Function	Unit
WA4	Normal	Black	Function	Unit
WA5	Normal	Black	Function	Unit
WA6	Normal	Black	Function	Unit
WA7	Normal	Black	Function	Unit
WA8	Normal	Black	Function	Unit
WA9	Normal	Black	Function	Unit
WA10	Normal	Black	Function	Unit
WA11	Normal	Black	Function	Unit
WA12	Normal	Black	Function	Unit
WA13	Normal	Black	Function	Unit
A1	Both	Black	Function	Unit
A2	Both	Black	Function	Unit
A3	Normal	Black	Function	Unit
FS1	Normal	Black	Function	Integration

FS2	Normal	Black	Performance	Integration