Group Name: Identfeye
Assignment: Test Plan
Team Members: Mitchell Haas, Zak Fahey, Brendan Fisher

Overall Test Plan

Our data needs to be tested in two ways. Firstly, we will use and test a data set logging the user behavior of a game server. We are working with this server to implement our project into their system. This data includes anonymized information including IP addresses, usernames, and whether the user has been banned. We can test using this data set as well as a manually constructed mock data set to ensure that the code works as expected. While we also want to analyze chat logs in this system, there is no practical way to safely anonymize this data, so it will not be included. We will instead test text analysis independently using a different data set so that we can ensure that it is functional. Once we do that, we can merge the text analysis into the main analysis code, and the third party developers (the admins of the server) can verify that it works correctly.

Test Cases

Test cases for server data

Server_1.1 Shared property trigger
Server_1.2 To verify that a data addition that includes a property that is shared by a banned user will trigger an alarm
Server_1.3 Data exists in system of a banned user. New data is entered into the system with one or more properties the same. The system should indicate that the new data point could belong to a banned user.
Server_1.4 Account name hash, character name hash, IP hash, IP geolocation hash, UUID hash, whether the user is banned, account age
Server_1.5 That the activity is associated with a banned user + the name of said user + the chain of association
Server_1.6 Normal
Server_1.7 Blackbox
Server_1.8 Functional
Server_1.9 Integration

Server_2.1 Shared property trigger 2
Server_2.2 To verify that a data addition that includes a property that has no association to a banned user will not trigger an alarm
Server_2.3 Data exists in system of a banned user. New data is entered into the system with no shared properties. The system should do nothing.
Server_2.4 Account name hash, character name hash, IP hash, IP geolocation hash, UUID hash, whether the user is banned, account age
Server_2.5 That the activity is normal

Server_2.6 Normal
Server_2.7 Blackbox
Server_2.8 Functional
Server_2.9 Integration

Server_3.1 Shared property trigger 3
Server_3.2 To verify that a data addition that includes a property that is shared by a banned user with one degree of separation will trigger an alarm
Server_3.3 Data exists in system of a banned user. Another data point includes a property shared by the banned user. A new data point shares a different property with the second data point. The system should indicate that the new data point could belong to a banned user.
Server_3.4 Account name hash, character name hash, IP hash, IP geolocation hash, UUID hash, whether the user is banned, account age
Server_3.5 That the activity is associated with a banned user + the name of said user + the chain of association
Server_3.6 Normal
Server_3.7 Blackbox
Server_3.8 Functional
Server_3.9 Integration

Server_4.1 Final third party test
Server_4.2 To verify that the program works in production
Server_4.3 System will be deployed on the third party game server and handle real data in real time. The third party will report back on the results of the test.
Server_4.4 Same as above, inputted whenever a player joins the server, sends a message, or logs in
Server_4.5 The system should send an alert successfully if and only if activity is detected from a user suspected to be associated with a banned account.
Server_4.6 Normal
Server_4.7 Blackbox
Server_4.8 Functional
Server_4.9 Integration

Server_5.1 Performance test
Server_5.2 To verify that the program works with large data sets at scale
Server_5.3 Test under the potentially 1 million+ data entries from our data set
Server_5.4 Account name hash, character name hash, IP hash, IP geolocation hash, UUID hash, whether the user is banned, account age. A list of items will be processed with this schema.
Server_5.5 The system should run the code in a reasonable amount of time with sub-O(N) time complexity.
Server_5.6 Normal
Server_5.7 Blackbox

Server_5.8 Performance
Server_5.9 Integration

Server_6.1 Account sharing alerts
Server_6.2 To verify that two active users with no prior link gaining a link triggers an alarm
Server_6.3 If there are two users who have a sufficiently large active playtime, i.e. they are both active players, and they have no prior link between each other, a sudden link between them could indicate that one member shared the account of the other.
Server_6.4 Account name hash, character name hash, IP hash, IP geolocation hash, UUID hash, whether the user is banned, account age.
Server_6.5 That the two users share an unexpected new link + the names of the two users
Server_6.6 Normal
Server_6.7 Blackbox
Server_6.8 Functional
Server_6.9 Integration

Test cases for text analysis

TA_1.1 Text Analysis Cleaned Data
TA_1.2 This will insure that the text analysis engine can handle cleaned up data in the form of an n-gram data set.
TA_1.3 For this test we create a model from the cleaned up data set and so if can feed the data through the model generator and have usable results.
TA_1.4 Inputs: cleaned up data set
TA_1.5 Outputs: model of data
TA_1.6 Normal
TA_1.7 Blackbox
TA_1.8 Functional
TA_1.9 Integration

TA_2.1 Text Analysis Raw Text
TA_2.2 To test if our project can text raw text and create the analysis text form and then feed it to the model generator.
TA_2.3 We will take a clean data set of new articles or chat conversations with labels, and create the analysis text then feed them into a model generator.
TA_2.4 Input: Data set of labeled text instances for multiple authors
TA_2.5 Output: Model
TA_2.6 Normal
TA_2.7 Blackbox
TA_2.8 Functional
TA_2.9 Integration

TA_3.1 Text Analysis Results
TA_3.2 This one is to test the model can give us usable predictions of whether or not unknown comments belong to someone in the model.
TA_3.3 We will use the rest of the data set that was set aside to see if any of the comments there match a known author.
TA_3.4 Input: unlabeled comments
TA_3.5 Output: percentage of confidence that a comment matches a known author
TA_3.6 Normal
TA_3.7 Whitebox
TA_3.8 Functional
TA_3.9 Unit

TA_4.1 Text Analysis Usability
TA_4.2 Test the integration of the results from the text analysis into the overall analysis that confirms if a user is a known person.
TA_4.3 Sending back the results of the text analysis to core analysis function as percent confidence, and a user name.
TA_4.4 Input: text comment
TA_4.5 Output: percent confidence, and a name or none found result
TA_4.6 Normal
TA_4.7 Blackbox
TA_4.8 Functional
TA_4.9 Integration

Test Case Matrix

ID	Normal / Abnormal	Blackbox / Whitebox	Functional / Performance	Unit / Integration
Server_1	Normal	Blackbox	Functional	Integration
Server_2	Normal	Blackbox	Functional	Integration
Server_3	Normal	Blackbox	Functional	Integration
Server_4	Normal	Blackbox	Functional	Integration
Server_5	Normal	Blackbox	Performance	Integration
Server_6	Normal	Blackbox	Functional	Integration
TA_1	Normal	Blackbox	Functional	Integration
TA_2	Normal	Blackbox	Functional	Integration

TA_3	Normal	Whitebox	Functional	Unit
TA_4	Normal	Blackbox	Functional	Integration