Identifeye

Zak Fahey, Mitchell Haas, Brendan Fisher

Purpose and Goal

- Changing your identity online can be as simple as flipping a switch.
- By using VPNs users can circumvent bans, or post as anyone online.
- People can also get into accounts who aren't supposed to access them.
- We plan to use data analysis to identify users who are changing their online identity.
- Essentially, we want to map online identities to real people, even if the data is obscured.
- We will achieve this by starting with a cleaned up dataset and discovering which attributes and classifiers work the best.

Who are we?

- Zak Fahey
  faheyzt@mail.uc.edu
- Mitchell Haas
  hassm3@mail.uc.edu
- Brendan Fisher
  fisherb8@mail.uc.edu
- Advisor: Raj Bhatnagar
  bhatnark@ucmail.uc.edu

Project Description

The object of this project is to build profiles on people and detect rule breaking behavior such as avoiding bans or sharing accounts.

Once we have identified these attributes we will test multiple analysis algorithms to find the optimal one for our analysis core.

After we have the analysis core setup we can apply it to many cases such as:

- mass duplicate account
- ban evasion
- account sharing

Computing and Software Design

- We will start with a cleaned up dataset that can be found here: data set
  - It is a set of Amazon Reviews
  - Each review is broken down into 1000 attributes that describe the authors linguistic style and use of digits, punctuations, words and sentence length, usage frequency of words, etc..
  - Their is 30 reviews per author so this data set can be used for authorship identification
  - We plan to add a few reviews of our to the dataset to have it classify when it is none of the known authors
- We will use this data set to test which classifiers work the best such as:
  - Neural Networks
  - Non-Linear SVM
- This process will help to identify which attributes work the best from the data set and which classifiers work the best for authorship identification.
- A stretch goal from here would be to then add natural language processing to convert plain text into our own data set and apply the best attributes and classifiers we discover.
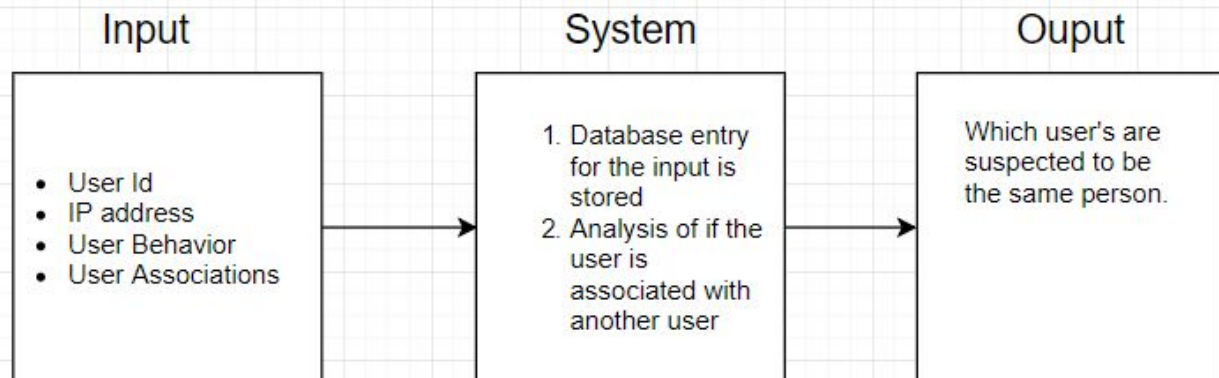
User Stories

As a developer, I want an easy-to-use, extensible, and capable user analysis program so that I can easily deal with the problem of users who abuse my system.
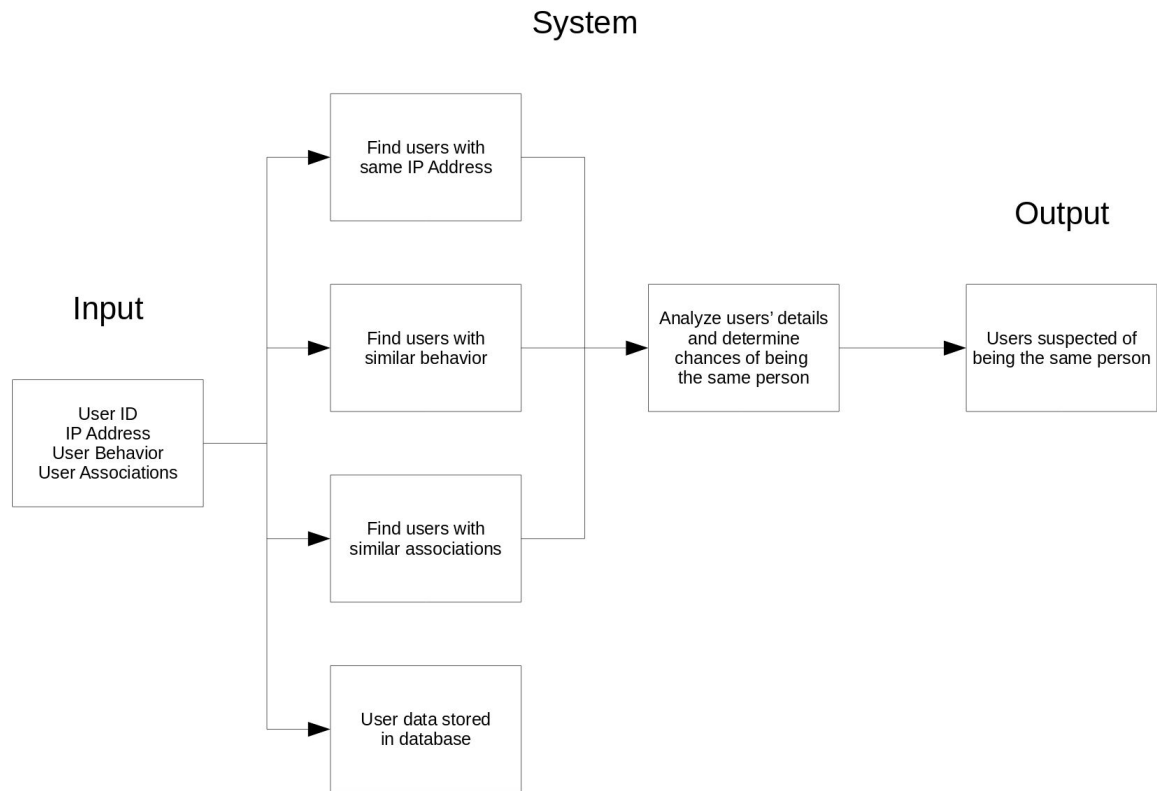
As an admin, I want a simple interface to visualize user trends so I can see where problems are and take action accordingly.

As a product owner, I want a solution that deals with abusive users so that my product is as good, useful, and free of problems as possible.

Design Diagram Level 0
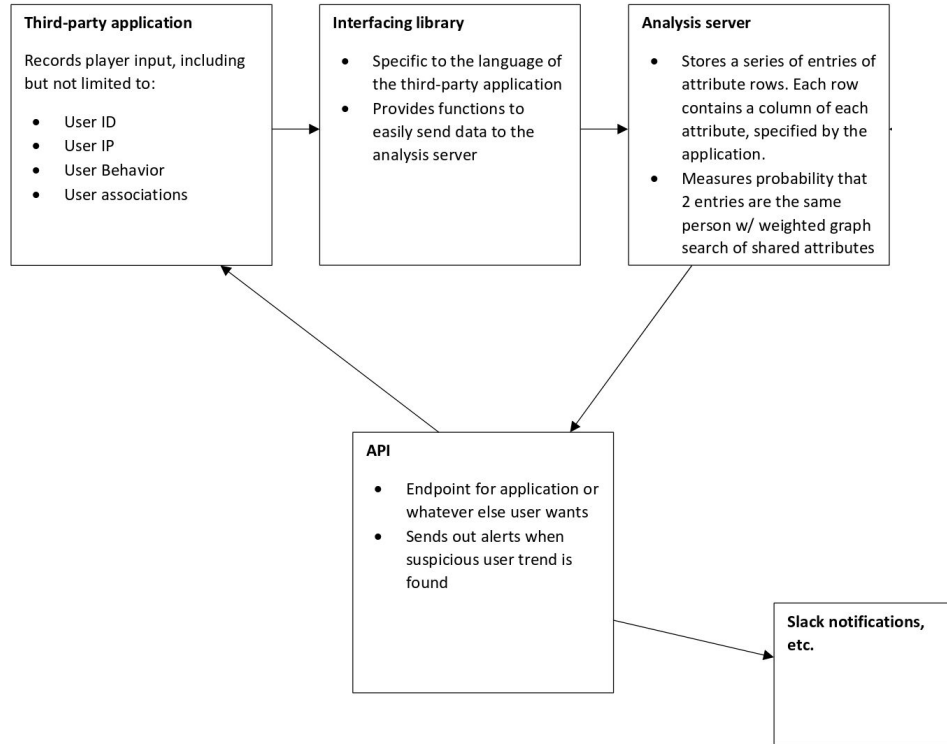
Input

- User Id
- IP address
- User Behavior
- User Associations

System

1. Database entry for the input is stored
2. Analysis of if the user is associated with another user

Ouput

Which user's are suspected to be the same person.

Design Diagram Level 2

Third-party application

Records player input, including but not limited to:

- User ID
- User IP
- User Behavior
- User associations

Interfacing library

- Specific to the language of the third-party application
- Provides functions to easily send data to the analysis server

Analysis server

- Stores a series of entries of attribute rows. Each row contains a column of each attribute, specified by the application.
- Measures probability that 2 entries are the same person w/ weighted graph search of shared attributes

API

- Endpoint for application or whatever else user wants
- Sends out alerts when suspicious user trend is found

Slack notifications, etc.

Review of Project Progress

- So far, we have completed the planning stage for the project.
    - Planning the division of labor
    - Finding a Data Set
- We have an understanding of how we're going to implement the code.
- Now all that's left to do is implement the code.

Division of Work

- Zak is leading the design of the core architecture of the systems. Pluggable library system, etc.
- Brendan and Mitchell are implementing specific subsystems for detecting specific behavior.

Task	Estimated Working Time in Hours	Group Members
Use gRPC to create an interfacing library in Python to communicate between the analysis server and clients	15 Hours	Zak
Develop the core of the analysis server in Python	60 Hours Total	(40 hours - Zak), (10 hours - Mitchell) , (10 hours - Brendan)
Find a cleaned up data set	5 hours	Mitchell
Code detection of account sharing.	15 Hours	Mitchell
Test the application with real-world data and verify that it works.	10 Hours	Mitchell
Find a classifier with the cleaned up data and test text analysis	20 hours	(10 hours - Mitchell), (10 hours - Brendan)
Code detection of mass duplicate accounts.	10 Hours	Brendan
Code detection of ban evasion.	10 Hours	Brendan
Create a flag for text content	10 Hours	Brendan