Identifeye

Team Members

Todo: add pictures

Zak Fahey Mitchell Hass

Brendan Fisher Dr. Raj Bhatnagar

Project overview

When running some online service, you have the problem of users attempting to hide their identity in order to avoid IP bans or log in to an account they shouldn't. Simply by using a VPN, users can change IP addresses on a whim, making it nearly impossible to know who a user really is. Our solution is to build a heuristics engine that can determine the real identities of users and whether they're doing anything malicious. We created this system to integrate with a gaming server for the game Terraria. It takes in data from the players such as their IP address and username and returns whether the user is doing anything suspicious. That data can then be used by the server to send alerts to staff about the suspicious behavior.

Data collection

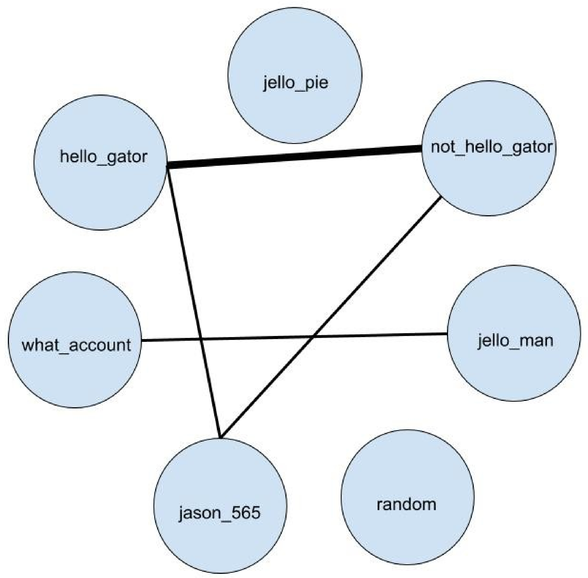
Our data was collected in partnership with Pedguin's Minigame Server, a game server that deals with malicious users at scale. We collected data on user accounts which included users' account names, character names, IP addresses, countries, UUIDs, number of minutes spent on the server, and whether they were banned. However, since this is sensitive data, we wrote a script that hashes all the values using the bcrypt algorithm. By using the same salt across all values, we can compare equality for values without knowing what those values are. There is still a salt, however, and it is not included in the data so that we cannot infer anything from known values by hashing it ourselves. However, our partners still have the salt, so they can analyze our results and cross-reference their own database to interpret our results.

Challenges

The hardest part of this project was figuring out how to architect the data analytics system and how to make it efficient. Also, basing our project off a third party made it more difficult to obtain our data and added a single point of failure if the third party decided to change their mind on giving away the data. Looking over and running the anonymizer program was considered a low priority for them, so it took quite a bit of time to receive the data.

Technologies Used:

ID	Account Name	Character Name	IP	UUID	IP Geolocation	Is Banned	Activity Time
0	hello_gator	Bob	127.1.0.1	56897	US	1	105
1	what_account	Jason	265.2.0.2	63541	Canada	0	212
2	jello_pie	Jason	315.6.4.7	88541	Mexico	0	105
3	jason_565	Rambo	127.1.0.1	11115	US	0	2
4	not_hello_gator	Bob	127.1.0.1	56897	US	0	5
5	random	Ryan	666.3.2.1	78912	Mexico	0	88
6	jello_man	Simba	111.5.6.0	63541	Canada	0	12



Sample data for testing detection of attempted ban evasion. Graph shows the different users and connects users which have similarities. Line thickness determines strength of the correlation between users.

Data analysis

The program collects a series of data points from any player interaction such as when they log in. A row is then added to our database that has all the associated properties for a user. When comparing one property to another, we can draw an association between two data points. The more properties are shared, the more likely it is that two points are the same person. You can then compare other nodes to that node and draw association through multiple degrees of separation.

To detect ban evaders, our database keeps track of a ban probability. A user who is actually banned has a ban probability of 1. When a new data point is added, all other data points are cross-referenced. For every shared attribute, the probability is increased up to whatever the old data point's ban probability is.

Components

Anonymizer - script to take source data and remove any sensitive information by hashing it
Interfacing library - C# library to communicate with the analysis server
Analysis server - main Python application to run data analysis