Behavioural Biometric Security

Ms Sneha Rose
Assistant. Professor CS & IT
Rathinam College of Arts and Science
Coimbatore, Tamil Nadu, India.
hello.sneharose@gmail.com

Ms Deepti R
B.Sc. Computer Science
Rathinam College of Arts and Science
Coimbatore, Tamil Nadu, India.
deepti05ramesh@gmail.com

Ms Kaaviya Sri S V
B.Sc. Computer Science
Rathinam College of Arts and Science
Coimbatore, Tamil Nadu, India.
kaaviyaselladurai@gmail.com

ABSTRACT

As e-Commerce continues and banking grows, shifting our shopping preference and money transferring from the physical to online marketplace, we leave behind digital traces of our personally identifiable details. For ex- ample, the merchant keeps record of your name and ad- dress; the payment processor stores your transaction details including account or card information, and every website you visit stores other information such as your device address and type. Cyber criminals keep robbing and use part of this information in fraud cases of identity, with tragic effects on the accountholders; but equally terrible to the card issuing bodies and payment processors, against whom financial liability most often falls. On the whole, we conclude that data is readily breached in this digital world and individual information like a credit card number, password, personal identification number and account details can be easily stolen and used by someone else. However, there is much data relating to a person's behaviour biometrics that are nearly impossible to steal, such as the way they type on a keyboard, move the cursor, or whether they normally do so via a mouse, touchpad or trackball. This data, commonly called keystroke, mouse and touchscreen dynamics, can be used to create a unique profile for the legitimate card owner, that can be utilised as an additional layer of user authentication during online card payments.

I.INTRODUCTION

An increase in online payments among the population has led to an increase in the identities of people and their personal details being vulnerable. Therefore, this project focuses on keeping the data of the users safe and continuously monitor the activities of the user to help enhance security. Here, the basic foundation is the use of a technology called “BEHAVIOURAL BIOMETRICS” to combat issue.

We aim to develop an application that acts as a platform to gets intricate details of a credible user that are difficult for a hacker to replicate. This web application captures and stores data such as Key Strokes, Movement of the mouse, etc. and monitors these activities continuously. In case of a, say, credit card details of a user being stolen, the hacker might have all the required information such as pin numbers and passwords, but the behavior of the credible users will be hard to replicate which will lead to the hacker being unsuccessful.

II. LITRATURE SURVEY

Several studies have highlighted the vulnerability of conventional security measures in combating identity theft. Research shows that static credentials (e.g., passwords, card numbers, and PINs) are often compromised due to phishing attacks, data breaches, and malware. Behavioral biometrics have been proposed as more secure alternative, as they rely on unique patterns of human interaction with devices

A. Machine Learning Applications:

Techniques such as Support Vector Machines (SVM), Decision Trees, and Neural Networks have been effectively used for pattern recognition in fields like bioinformatics and cybersecurity. For example, genome classification relies on machine learning to identify subtle patterns in complex datasets, which is analogous to detecting anomalies in user behavior during online transactions.

B. Financial Sector Applications:

Although behavioral biometrics are well-studied in other domains, their application to prevent online payment fraud remains underexplored. This gap presents an opportunity to integrate these techniques into financial systems to enhance security.

C. Digital Data:

As online payments have become integral, personal data such as names, addresses, and payment details are stored by merchants and processors, making them vulnerable to cybercriminals and identity fraud.

D. Impact of Cybercrime:

Stolen data can lead to severe consequences for victims, card issuers, and payment processors, who often bear the financial liability.

E. Behavioral Biometrics:

Data on keystroke, mouse, and touchscreen dynamics, (ie), how a person types, moves the cursor, and uses input devices—are unique and hard to steal, offering a potential layer of security. There are a number of solutions available in the tech market but the solutions too lack in few areas. Mainstream transaction applications such as “Google Pay” and “Paytm” are not very secure either.

F. End Goal:

The project aims to provide security for the data of the users with continuous monitoring. This project not only prevents any sort of crimes from happening , but also aims to keep the data safe from hackers in case of a data compromise. In case of a, say, credit card details of a user being stolen, the hacker might have all the required information such as pin numbers and passwords, but the behavior of the credible users will be had to replicate which lead to the hacker being unsuccessful. Though the project for now is extremely credible, it does lack in a few areas. Here, we build an application that not only monitors the user’s activities but also sends notifications to the user.

But this is where the lack is. The application sends notifications to the user but fails to let the user know who could be the possible potential hacker.
Therefore, our web application has an additional feature where as soon as an unusual activity is sensed, the camera automatically captures picture of the person in control of the activity at the moment. This helps in not only help the user know the hacker but also helps prevent the guilty from doing any further crimes.This also helps the user to not misjudge a family member or a loved with that has the credentials of the user with the user’s consent as a potential hacker.

III. PROBLEM STATEMENT

The increasing prevalence of identity theft and financial fraud highlights the inadequacy of traditional authentication systems for online transactions. These methods, based on static credentials, are easily compromised, leaving users and financial institutions vulnerable. A real-time, behavior-based authentication mechanism is needed to address this gap and protect sensitive user information during online card payments.

IV. PROPOSED WORK

This research proposes the development of a JavaScript-based system that leverages behavioral biometrics for online transaction authentication. The key steps include:

A. Data Collection:

The project, as said before, aims on collecting the user behavioral biometrics data and using the collected data in the money transaction process.
Here, the web application that we aim on developing begins with giving a guide tour to the user. This lets the user know how the application operates.
Next up, to collect the actual behavioral data the application provides the user with a 120 words paragraph and allows the user to type the paragraph in their own phase and speed.
The maximum time allotted for this process is 3 minutes at the maximum.

B. Data Storage:

Once the user finishes typing in the paragraph, the collected data gets stored in a database.

The database will also contain other information of each respective user, such as mobile number, email id etc. In this application, as said before, if and only if all the credentials including the pin number, password (if any), and the behavioral biometrics match with the credentials of the original user , will the transaction proceed.

C. Behavior Matching:

This web application will match the behaviors of the user using the credentials with the ones stored in the database. The currents speed, pace, rhythm etc. of the user should have to match above 85% with the data stored in the database. This web application doesn't wait for a 100% match, since the typing of even the authorized user might differ. Therefore, a match of 85% and above seems legitimate.

D. Combatting The Issue:

In case of the user activity not reaching the ideal percentage of matching, the application will flag the transaction. Once an unusual activity is sensed , the application stops the transaction midway and by default, triggers the camera. This operation will capture the user who is in control of the application and sends it as a notification to the original user. This will help the user to know was the person in charge of the unusual activity and also prove to be an evidence in case of a legal issue

E. Notifying The user:

The notification sent will be an alarm notification and the notification with the picture will be sent through mail. When the user is notified of the unusual activity, the user will be provided with two options, either to "Ignore" or to "Proceed Transaction" . The "Proceed Transaction" is a necessary feature because if the transaction is an emergency and is happening with the consent of the user, the transaction will proceed further. If the user chooses to "Report" the notification, the transaction that was paused midway will be suspended or dismissed. Though the user chooses to "Report" the notification, the user will still have the picture of the person responsible for the activity and either way, the transaction will be suspended resulting in the user's money being safe.

V. TOOLS REQUIRED:

Frontend Tools:

- React.js – For building the frontend UI.
- Webcam.js – To capture images from the user's webcam.
- HTML, CSS, JavaScript – Core web technologies for styling and interactivity.
- Axios or Fetch API – To communicate with the backend.
- WebSockets (socket.io-client) – For real-time alerts and notifications.
- React Hooks (useRef, useState, useEffect) – For state management and capturing images.
- React Router – For navigation between multiple pages.
- Google Material UI (or Tailwind CSS) –
- JWT (JSON Web Token) in LocalStorage – To store authentication tokens securely.

Backend Tools:

- Node.js + Express.js – To build the backend API.

- MongoDB + Mongoose – For storing user biometric data securely.
- WebSockets (socket.io) – For real-time alerts.
- Nodemailer – To send email alerts for suspicious activities.
- FS (File System) Module – To save captured images on the server before emailing.
- Dotenv – To manage environment variables securely (.env file for credentials).
- CORS Middleware – To allow communication between frontend and backend
- JSON Web Token (JWT) – For secure API authentication.
- Bcrypt.js – For hashing passwords securely.
- Morgan – For logging API requests (if used for debugging).
- PM2 – To keep the Node.js server running in production.

Security Features:

- Behavioral Analytics class:
- Mouse movement tracking
- Keystroke pattern analysis
- Pattern complexity calculation
- Confidence scoring

Camera Integration:

- Webcam access and control
- Image capture
- Base64 image conversion

Email Service:

- Security alert management
- Email notifications
- Alert status tracking

Development Tools:

Frontend:

- React.js
- HTML , CSS , JavaScript
- React Webcam
- Axios / Fetch API
- React Route
- Redux / context API

Backend:

- Node.js
- Express.js
- Mongoose
- Nodemailer
- Socket.io
- Bcrypt.js
- JWT
- DOTENV

Core functionality:

- User Input Management
- Email field tracking
- Password field tracking
- State management using React hooks
- Real-time behavioral data collection

Behavioral Analysis Integration:

- Tracks mouse movements
- Monitors keyboard patterns
- Calculates behavioral scores
- Triggers security measures

Security Camera Integration:

- Initializes camera on component mount
- Captures images on suspicious activity
- Manages camera permissions
- Handles image processing

Alert Management:

- Displays security alerts
- Manages alert responses
- Handles alert status updates
- Coordinates with email service

Key Methods:

Handle Submit:

- Prevents form default submission

- Triggers behavioral analysis
- Manages authentication flow
- Handles security measures

Handle Suspicious Activity:

- Captures camera image
- Triggers email alerts
- Updates security status

Handle Authentication:

- Processes login attempts
- Validates credentials
- Manages session state
- Updates user status

Security Alert Component (SecurityAlert.jsx)
Purpose:

- Displays security notifications
- Provides user action options
- Manages alert responses
- Maintains UI consistency

Features:

- Modal Interface
- Overlay background
- Centered alert box
- Responsive design
- Accessible controls
- Proceed Buttons
- "Report" option
- "Proceed" option
- Loading state handling
- Event propagation

Pattern Analysis:

- Velocity calculations
- Movement complexity
- Rhythm consistency
- Confidence scoring
- Keyboard Analysis
- Keystroke timing
- Typing patterns
- Rhythm analysis
- Pattern matching

Analyze Behavior:

- Combines mouse and keyboard data
- Calculates confidence scores
- Determines authenticity

- Provides detailed metrics

Calculate Pattern Complexity:

- Analyzes movement patterns
- Determines user consistency
- Evaluates behavior uniqueness
- Generates complexity scores
- Camera Integration (WebcamCapture.js)

Main Features:

- Camera Management
- Image Capture
- Frame extraction
- Image processing
- Format conversion
- Quality management

Resource Management:

- Stream handling
- Memory cleanup
- Error handling
- Device permissions
- Email Service (emailService.js

Functionality:

- Alert Management
- Database storage
- Email notifications
- Status tracking
- Session management

VI. RESULT ANALYSIS

Enhanced Security:

The users, with the help of this web application will be able to stay safe from transaction hacks and loss of money and personal information. Here, the application not only notifies the user but also makes sure to trigger the camera and capture a picture of the person in control of the transaction at the moment which could help the legitimate user as an evidence in instances of a legal issue.

Reduced False Negatives:

False negatives are often an issue when it comes to pattern matching. Therefore, this web application operates efficiently where a match equal and above 85% becomes credible.

Dynamic Build:

The web application is dynamically built, making the application safe and at the same time extremely efficient. The application is designed to flag the unusual activity the moment it is sensed and notify the user at the earliest. But at the same time, if the user chooses to let the transaction happen despite the warning, the transaction will smoothly proceed instead of getting suspended.

VII. CONCLUSION

The integration of behavioral biometrics, such as keystroke and mouse dynamics, offers a robust solution to the growing challenges of identity theft and financial fraud in online transactions. By leveraging unique user interaction patterns, this approach enhances security beyond traditional methods like passwords and card numbers, which are vulnerable to compromise. The application of machine learning techniques to analyze these patterns has demonstrated significant potential in distinguishing between legitimate and illegitimate users with high accuracy.

This study provides a foundation for developing real-time authentication systems that can effectively mitigate risks associated with unauthorized transactions. The collected dataset and preliminary results indicate that behavioral biometrics can serve as a reliable, non-intrusive additional layer of security. Future work could explore expanding the dataset, incorporating additional biometric features, and optimizing machine learning models to further enhance system performance and adaptability.

VIII. REFERENCES

[1] The European Parliament and of the Council, Directive (EU) 2015/2366 of the European Parliament and of the Council (2015). URL https://eur- lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32015L2366 &from=EN .

[2] N. Nnamoko, J. Barrowclough, M. Liptrott, I. Korkontzelos, Behaviour Biometrics Dataset, Mendeley Data (2022) v1 URL https://data.mendeley.com/datasets/fnf8b85kr6 , doi: 10.17632/fnf8b85kr6.1 .

[3] I. Guyon, A. Elisseeff, An Introduction to Feature Extraction, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–25. doi: 10.1007/978- 3-540- 35488- 8 _ 1 . URL https://doi.org/10.1007/978- 3- 540- 35488- 8 _ 1 .

[4] N. Nnamoko , F. Arshad , D. England , J. Vora , J. Norman , Evaluation of Filter and Wrapper Methods for Feature Selection in Supervised Machine Learning, in: PGNET, 2014, pp. 63–67 .

[5] R.J. Quinlan , C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993 .