Keystroke Dynamics Based Authentication As A Service

Rushin Tilva, Akshay naik, Dr. Ambika Pawar, Rupali Gangarde

Abstract: With the increase in use of internet, there is an increase in vulnerabilities and security lapses which lead to theft of sensitive data has also increased. To protect data, password-based authentication systems are used. But they are still vulnerable if the user as set one of the very common passwords which can be easily cracked by simple guessing or can be stolen or copied if you wrote the password somewhere and lost it. For better security a second layer of authentication like finger scan are used but need special hardware and become costly to implement.

Index Terms: authentication, anomaly detection, dynamics, keystrokes, password, vulnerable

1. INTRODUCTION

THE combination of username and password has been the old method of authentication into any system. This is reliable until the password is not compromised in a password database leak or is lost and falls in hands of a wrong person or by simple password guessing. This lapse of security can lead of loss of sensitive data of the user or unauthorized actions be performed under the name of the hacked user’s account. Multiple techniques exist which can be implemented to tackle this problem. First method which is commonly used is setting some criteria for accepting the username-password combination during sign up. These criteria would mention the minimum password length, compulsory inclusion of special characters other than usual alphanumeric characters. This strategy surely protects against simple password guessing as well as password brute-forcing to an extent but doesn’t secure against stolen password. Another approach to securing the password is by using an biometric identity like fingerprint, iris scan, etc. But this approach has a major drawback of requiring the scanner device to scan the fingerprint or iris, every time the user wants to login. Another approach is implementation of second layer of authentication like OTP, QR codes, time-based password (TOTP) etc. Only after the user clears the first layer of authentication which is, his password matches the one stored in the database, he is challenged with the second layer which could be one of the mentioned above. Like in the OTP based authentication, the user is asked to enter the OTP sent to his/her mobile phone and only then is considered authenticated. These methods are slightly inconvenient as they have constraints like needing the mobile phone handy. Keystrokes dynamics-based authentications is type of a second layer of authentication which does not reduce the convenience of the user. It naturally extracts the typing rhythms from the user’s typing of his password during the first layer. This method can uniquely identify various users even when both type the same password to authenticate. This method of authentication can be applied anywhere where a user would use a normal keyboard to type his password or a phrase. This would include password fields, credit card numbers, email address, user names etc. This method of biometric authentication can also be implemented for continuous typing like while writing an essay. In this scope of research, we propose a two-factor authentication with password as the first factor implemented on the server side and keystroke dynamics as the second factor implemented as a standalone service. The system aims to obtain the minimum criteria of 0% False Acceptance Rate (FAR). We also discuss ways of archiving optimal accuracy as well as maintaining this accuracy over a period.

2 INTRODUCTION

In most of the previous research, a particular subject is classified as an imposter or a legit user with the help of factors like transition time between two keys (time interval between the two consecutive key presses) and hold time (time interval between key press and release of a specific key) of keys when user types the password. Keystrokes dynamics based authentication framework was made using three-layered back propagation neural network having adaptable inputs for password, by setting convergence criteria, root mean square error (RMSE) to a lesser threshold value during neural network training process it gives 1% false rejection rate and 0% false acceptance rate[1]. But as the time passes and user gets more familiar with typing the same password for many times, the typing pattern of user changes (speed increases) which was not accommodated. A more better solution was given by Kevin S. Killourhy and Roy where they collected data of 51 subjects typing 400 times the same password and compared 14 detectors from keystrokes dynamics literature. Using that data they concluded that Manhattan (scaled) detector gives the best results[2]. Now to accommodate the change in typing pattern of user over time can be done by sliding window or growing window methods mentioned [3] by Pilsung Kang where only latest few keystrokes dynamics of the user are used to retrain the model [3]. There are some papers which discuss free text system for continuous authentication out of which Arwa Alsultan and Kevin Warwick has discussed a solution which provides proper balance between security and usability in the applications where continuous authentication is needed[4]. A web based keystrokes dynamics based authentication system [5] was discussed by Sungzoon Cho, where auto associator neural network was trained with the data of owner and then with the impostor. The impostor

---

• Rushin Tilva is currently pursuing Btech computer science in Symbiosis International University, India, E-mail: rushintilva@gmail.com
• Akshay Naik is currently pursuing Btech computer science in Symbiosis International University, India, E-mail: akshay.naik@sitpune.edu.in
• Dr. Ambika Pawar Associate Professor in Symbiosis International University, India, E-mail: ambikap@sitpune.edu.in
• Rupali Gangarde Assistant Professor in Symbiosis International University, India, E-mail: rupali.gangarde@sitpune.edu.in
was detected with high accuracy and this model can be implemented by java applet also. Another research done by Sungzoon Cho concludes that touch pressure, touch size and coordinates are also good features to identify the user, they have used RMR feature selection to increase the classification performance matrices. They have used support vector machine classifier and multi class classification with one vs. one decision shape function[6]. The Pawel Kobojeck and Khalid Saeed has proposed system which propose recurrent neural networks for user verification based on keystrokes dynamics. They have used LSTM and GRU and achieved good results in reducing false positive while authenticating the user. The system works with sequential data input and perform with different architectures [7]. The MultiLock: Mobile Active Authentication based on Multiple Biometric and Behavioral Patterns was propose by Alejandro Acien [8] where they have worked on mobile active authentication and behavior profiling signal. They have used seven different data channels namely keystroke dynamics, touch gesture, accelerometer, gyroscope, WiFi, GPS location and app usage. Evaluated two approaches active authentication and one-time authentication. Experiment is done on semi-uncontrolled UMDAA-02 database. Performance with accuracy from 82.2% to 97.1% in different scenarios is done [8]. Keystroke Dynamics for Mobile Phones: A Survey by Baljit Singh Saini [9] specifies that latency, hold time and pressure are used as features in most of the papers. According to the research most papers used statistical or neural network. Most of the papers have used less then 100 people for testing which is bad according to author [9].

The Keystroke dynamics: Characteristics and opportunities survey done by Heather Crawford specifies a united method to compare different research done on keystrokes dynamics based authentication systems. They have compared few research papers and provided future scope along with set of guidelines which can be followed by researchers willing to do future work on KDBAS.

3 IMPLEMENTATION

Fig. 1. shows system architecture of proposed system.

3.1 BASE MODEL

Because The base model implemented is the Scaled Manhattan distance based anomaly Detection model. During the first registration, the user is made to enter is password 5 times. These entries will be used as training samples to train the model. The features recorded during this stages are hold time, seek time, char code (ASCII value) and the keyboard key code for each keystroke. Then the mean vector of all the 5 attempts is calculated. Then this mean vector is used to calculate the individual Manhattan distances with each of the 5 training samples. The max distance among all the samples is recorded and is set as threshold.

3.2 LOGIN PHASE

The first step of the authentication system would be to verify if the entered password matches the password in the database. The inputted password is hashed using a predefined hash function. The user is searched in the database using his email and corresponding password hash is fetched from the database. Both the hashes are compared, if the hashes match then the keystrokes dynamics are checked in the next step. During the login phase the same features mentioned above are recorded and will be used as the test vector. If the password matches, the Manhattan distance is calculated between the test vector and the mean vector which was calculated in the registration step. This distance is compared with the threshold. If the distance is less than the threshold then we consider the authenticator as legal else is rejected as an imposter.

3.2 CONTINUOUS LEARNING

It is necessary to keep updating the user’s keystroke dynamics’ bio metrics over a period. Asking the user often to explicitly register his keystroke is not convenient nor seamless. This problem can be solved with the proposed solution of continuous learning. Keystrokes dynamics data can be updated implicitly on the event of successful login. This will add the user’s latest biometric (which is slightly changed compared to last recorded) data to the database and recalculate the mean vector and threshold.

3.3 SLIDING WINDOW SAMPLING

It is well observed that the keystroke dynamics of a user change over time and never remain the exact constant. Hence a model once trained and used forever is bound the fail over a period. The proposed solution to this is using only the last 5 samples recorded and calculating the mean vector and threshold from them.

3.4 SERVICE

Any developer must be able to easily deploy this architecture into his own system, hence the keystroke dynamics service exposes two endpoints. The first endpoint is /addData which is used during registration of the user. It accepts an email and 5 keystroke dynamics data strings. The second endpoint is the /predict endpoint. This takes input a single email and a single keystroke dynamics data string. It validates if the user is genuine or an imposter and returns the result. All the code base files are packaged in a single easily deploy-able package.
4 RESULTS
In this section we will discuss the results we obtained from our performed experiments. The following experiments were performed on the above-mentioned benchmark data-set.

4.1 PLOTTING TIMING VECTORS
The working principle of our algorithm majorly relies on segregating the users based on the time they spend while interacting with a particular key or key pair while typing their passwords. As shown in Table 1, in proposed experiments the timing data of 2 randomly selected typist was plotted on a single graph. In this case we selected data of typist ‘s002’ marked in colour red, ‘s007’ (sessionId 1-3) marked in color green and ‘s007’ (SessionId 6-8) marked in colour blue. Along with this a solid line of the respective colour is all plotted for each subject which signifies the median timing of the typist. The data of typist s002(red) and s007(green) are from their sessions 1 to 3 while data of typist s007(blue) is from sessions 6 to 8. By plotting the data in this way, we can see that compared to the red and green lines, blue line occurs much lower than the other lines which is of typist's early typing data. Also, we calculated the standard deviations of the 3 subjects and observed that blue has a much lower standard deviation. From this we can infer that over a period of time the average timings reduce for a particular user and also his variations in typing reduce and a more consistent typing is achieved. Exception to this we see is the hold timing vector which was not much affected over a period of time.

<table>
<thead>
<tr>
<th></th>
<th>H.period</th>
<th>DD.period.t</th>
<th>UD.period.t</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>s002</td>
<td>0.0132</td>
<td>0.066</td>
</tr>
<tr>
<td>Green</td>
<td>s007 (S1-3)</td>
<td>0.0147</td>
<td>0.103</td>
</tr>
<tr>
<td>Blue</td>
<td>s007 (S6-8)</td>
<td>0.0119</td>
<td>0.023</td>
</tr>
</tbody>
</table>

4.2 HOLD TIME TIMING VECTOR
Hold time is the time elapsed while the key is kept pressed. This feature has the lowest variance among other features that we compared, making it the most useful feature among available. Hence, we will be using it while making our final model. Fig. 2. shows hold time timing vector.

4.3 DOWN-DOWN TIME TIMING VECTOR
Down-Down time is the time between 2 key-down events of 2 continuous password characters. While initially (Green) it had a high variance, in the later stages a lower variance was obtained and hence we will be including it for making the final model. Fig. 3. shows Up-Down time timing vector.
4.4 FEATURES IMPLEMENTED
Fig. 4. Shows following features used in making of the models of various detectors.

<table>
<thead>
<tr>
<th>Author</th>
<th>Detector</th>
<th>Hold Time</th>
<th>Key Down-Down Time</th>
<th>Enter Key Up-Down Time</th>
<th>Samples Moving Filtering Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joyce &amp; Gupta (1990)</td>
<td>Manhattan (Filtered)</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Araujo et al. (2004)</td>
<td>Manhattan (Scaled)</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cho et al. (2000)</td>
<td>Neural Network (auto-assoc)</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Haider et al. (2000)</td>
<td>Neural Network</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>This Paper</td>
<td>Manhattan (Filtered)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neural Network</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neuron Network (auto-assoc)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.5 DETECTOR EVALUATION (EQUAL ERROR RATE)
In this study we will be using the equal-error rate to evaluate our detectors against each other. The equal error rate is the rate at which both the zero-miss rate and false-alarm rate are equal. The least(optimal) equal error rate was obtained by the model proposed by this paper, which was 0.05.

4.6 EER OBTAINED COMPARISON
Table 2. shows EFR comparisons of past papers.

<table>
<thead>
<tr>
<th>Author</th>
<th>Detector</th>
<th>Zero-miss Error Rate</th>
<th>Accuracy with 0% FAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joyce &amp; Gupta (1990)</td>
<td>Manhattan (Filtered)</td>
<td>0.22</td>
<td>78%</td>
</tr>
<tr>
<td>Araujo et al. (2004)</td>
<td>Manhattan (Scaled)</td>
<td>0.601</td>
<td>39.9%</td>
</tr>
<tr>
<td>Cho et al. (2000)</td>
<td>Neural Network (auto-assoc)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Haider et al. (2000)</td>
<td>Neural Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This Paper</td>
<td>Manhattan (Filtered) + Neural Network + moving window</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

Comparison of Manhattan (Scaled) Model with and without using Moving Window Protocol
The result shows a greatly improved zero-miss error rate and accuracy by using moving window protocol. While using moving window protocol the model correctly classified 78% of the samples without accepting a single imposter (0% FAR). The previous works could only obtain an accuracy of 39.9%.

4.7 FAR SCORE WITH MOVING WINDOW
Following table shows standard deviation of features compared of various subjects across different time frames. Result shows Manhattan (Scaled) with moving window shows acceptable results.

CONCLUSION
Consolidating the keystroke dynamics with the typed password is a substantially compelling approach to validate the legal access because of the special keystroke characteristics of every person. In view of the process that clients have unique typing rhythms, this authentication system provides efficient and affordable online validation web framework by using anomaly detection.

ACKNOWLEDGMENT
We would like to thank CMU for providing a cleaned keystroke data set which could be used for testing our model

REFERENCES


