Computer User Verification Based on Mouse Activity Analysis

Tomasz Wesolowski, Malgorzata Palys, and Przemyslaw Kudlacik

University of Silesia, Institute of Computer Science,
Bedzinska 39, 41-200 Sosnowiec, Poland
{tomasz.wesolowski,malgorzata.palys,
przemyslaw.kudlacik}@us.edu.pl

Abstract. The article concerns behavioral biometrics, where issues related to computer user verification based on analysis of the mouse activity in computer system are particularly studied. The work is devoted to the analysis of user activity in environments with a graphical user interface (GUI). The set of analyzed features is extended by introducing new features to characterize the mouse activity. A new method of aggregating the mouse activity basic events into a higher level events is presented. Additionally, an attempt to intrusion detection based on the *k*-NN classifier and various distance metrics is performed.

This article presents the preliminary research and conclusions.

Keywords: behavioral biometrics, mouse activity, user verification.

1 Introduction

The computers are present in almost every aspect of our life and the security of the computer systems and data became a crucial task. People use computers to perform calculations, manage finances, support health care [10,17] and rescue services or simply for entertainment. Institutions often process and store confidential information. Among them financial documents, projects documentations or personal and sensitive data of the users. Therefore, it is necessary to take appropriate security measures for the IT systems that will allow very high level of access control and only the authorized individuals should gain access to the specific resources. In order to allow such a security measures it is necessary to confirm the identity of the user. To increase the security level advanced methods are expected, e.g. biometric methods that identify users by their individual physical (face, fingerprint) or behavioral (signature [9], walking style) characteristics. Also using the peripherals of the computer (keyboard, mouse) is considered to be the human behavioral characteristics. Unfortunately the analysis of the keyboard use as a continuous process is very difficult due to the fear that the personal data (passwords, PIN codes) could leak or get stolen. This is why mouse activity, as more safe for the user, should be considered separately. Another important reason for analyzing mouse activity is that this approach is not involved with any additional costs (no sensors or devices are necessary to

D. Barbucha et al. (eds.), *New Trends in Intelligent Information and Database Systems*,
Studies in Computational Intelligence 598, DOI: 10.1007/978-3-319-16211-9_7

collect the data). The data acquisition methods used by the authors in these studies could be easily implemented in the security software like the host based intrusion detection system (HIDS).

When using a computer and an operating system (OS) with the graphical user interface (GUI) a user almost continuously works with computer mouse. This is how an individual style of interaction is developed. The assumption for using the mouse activity in biometrics is that this individual style could be unique for a user and could characterize this user in order to perform a user verification. Such an interaction could be very difficult to falsify because it is developed in a long period of time and depends on individual personal features of a user. At the time the mouse movements are already used in user authorization (image passwords). Unfortunately there is a major issue connected to these authorization methods - they are one-time operations usually performed at the beginning of work. The one-time authorization does not guarantee that an intruder will not overtake the access to the system where the authorized user is already logged in - this can happen when the authorized user leaves for some time the working place after logging in into the system. Such an intruder is called a masquerader and usually it is an insider. Within different kinds of cyber-attacks [12] masqueraders constitute the greatest threat [13]. To ensure a high level of security the activity of the user should be continuously monitored and the working person should be verified. Such a monitoring is performed by the IDS. Users' profiles are created and by means of various methods the reference profiles are compared with the users' activity data. If such a comparison is made on the fly (in real time) it is an on-line IDS (dynamic analysis methods are used) and if it's made after some time the IDS works off-line (static analysis methods are used).

The user profiling and intrusion detection method presented in this article were developed as dynamic to work with an on-line IDS to monitor the activity and verify the user in a real time. The presented approach represents the preliminary studies. The studies apply for the OS with GUI and the activity data is collected on the OS level so all the computer programs with GUI are covered (also web browsers, working with web pages and social media).

2 Related Works

The idea of user verification based on the activity in a computer system is well known in behavioral biometrics. At first, the interacting with the computer system was based on giving text commands. These commands were recorded and analyzed in order to create an individual user profile [14,15]. Another approach consisted of the direct use of the keyboard analysis [3]. Along with the development of computer systems due to the increasing computing power the way of human-computer interaction also has evolved. Nowadays, mostly OSs with GUI are used and some solutions combine the mouse and keyboard activity [1,16].

This paper is focused on using the information on computer mouse activity only. Some of the methods are developed to assist during a one-time logging in authorization. In [5] the mouse activity data set was acquired through a game placed on the internet website. The players were assigned individual id numbers and the game was collecting the information on mouse cursor position. As a classifier a Sequential Forward Selection algorithm was used. The methods for continuous analysis of the activity are more difficult to develop but they can be used in an on-line IDSs. The entertainment business has a big influence in this area because of the need to look for new methods to secure the computer games - especially the Massively Multiplayer Online games. These games are a part of a complex economical and financial mechanism. The personal and financial (credit card numbers) data of the users are stored. The analysis of mouse activity in games in order to identify the users was made in [7,8]. In [7] the trajectory of the mouse cursor was analyzed by using the signature verification methods, SVN and k-NN classification. Another solution is based on the separation of mouse events (move, click) and their subsequent analysis [11]. The distance, angle and speed of movement of the cursor was calculated and then the average values were determined. Gross deviations from the calculated average values consisted anomalies that were classified as the presence of an intruder. The analysis of continuous user activity was performed in [2] where the dynamics of mouse use based on the moves characteristics is presented. The activity data was collected during users' everyday activity. Basing on the data features were extracted and histograms characterizing individual users were depicted. Based on [2] further improvement was introduced in [4] where some additional features were analyzed and the Equal Error Rate (EER) was lowered in average of 1.01%. Other approach was based on calculating the vector of characteristic features [6]. The calculated vector includes information about the average number of the left and right mouse button clicks and a thorough analysis of the trajectory of the cursor. The effectiveness in verifying the identity of the person was around 80%.

In this paper the set of features is extended by proposing new features. Also a new way of dividing the mouse activity into a higher level events is presented.

3 Testing Data Set

The data set used in this studies was collected by the dedicated software. The software works in the background of the OS registering user activity at every moment of his work. Despite this, the application is unnoticeable to the user and does not affect the comfort of his work. The users were of various age, occupation and level of computer use experience. Each user was working on his/her own personal computer. The software records activity saving it in files compatible with the CSV format. The first line starts with a token RES followed by the resolution of the screen. Starting with the second line the activity data is stored and it consists of mouse and keyboard events. The software recorded even 60000 events per hour. Due to the security reasons the key codes and window titles are encoded. Each line starts with a prefix identifying the type of event. The prefix is always followed by the timestamp and optionally additional data related to the event. The prefixes for mouse activity events are: M - mouse move, L/l - left mouse button down/up, R/r - right mouse button down/up, S - mouse scroll.

Each mouse event consists of a vector [*prefix*, *timestamp*, x, y] where x and y are the coordinates of the mouse. The negative values of coordinates are possible when working with multiple screens. Despite the fact that the keyboard codes and window names are encoded the data acquisition process is very difficult as most of the users are not willing to voluntarily participate due to the fear of the personal data leaks. Initially the data from 10 users were collected but the software is still running to extend the testing data set and collect more data.

3.1 Data Pre-processing – Higher Level Event

As the acquired data set consists of activity data connected to mouse, keyboard use and activity in particular software/windows the pre-processing is necessary. The first step is to filter the data set in order to acquire only the mouse basic events (moves and button clicks). Next step is to extract such a data subsets that are connected to a mouse activity from which the features could be extracted.

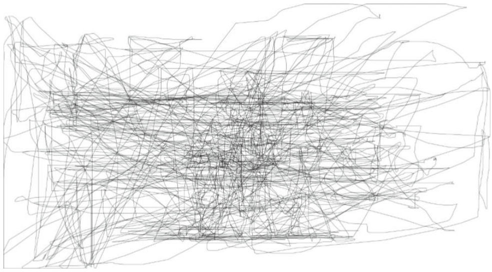

Fig. 1. An example of mouse cursor path after an hour of work

After working for a longer period of time the mouse activity data have a very high number of basic events (Fig. 1). Extracting the features from such a volume of data is complex and time consuming. Therefore it is necessary to divide the mouse activity into sequences of basic events that constitute the *higher level event* (*HL-Event*) and then analyze the HL-Events separately. For the purpose of data analysis in this studies the HL-Event consists of all basic events that were recorded between two consecutive clicks of the mouse button (Fig. 2b). Because the activity in the close neighborhood of the click event and the click event itself could be important also these activities are analyzed separately.

There is some issues related to data pre-processing. The first problem is that the data recording software saves a double-click event as two single clicks, therefore it is necessary to extract double-click events during the pre-processing. Two single clicks are interpreted as a double-click when the time difference between them was less than 0.5 second and the distance less than 5 pixels. Next the double-click is interpreted as the border event for HL-Event. Another issue is related to Drag-and-Drop operations (DaD). The DaD consists of mouse button down event followed by one or more mouse move events and ended with mouse button up event. It is similar to the regular HL-Event where the border events are mouse clicks. This is why it is interpreted as HL-Event at the end of which instead of the mouse click the mouse button up event is registered. Next problem is the mouse inactivity - when the mouse cursor stops at the particular place for some time. The situation like this takes place for example when the user is: waiting for the software, reading or typing something on the keyboard. Such a situation has a negative influence on the features describing dynamics of the move. To eliminate the disadvantage the basic events till the mouse inactivity point are ignored and the basic events between inactivity and next mouse click constitute the HL-Event.

4 Mouse Activity Features

Every mouse move basic event is characterized by the vector of 3 values $[t, x, y]$ consisting of the horizontal (x) and vertical (y) coordinates of the mouse cursor in the time t. The HL-Event consists of $n + 1$ basic events and hence is a set of $n + 1$ such vectors. The value of n represents at the same time the number of points of the trajectory - without the initial point number 0. Therefore, the HL-Event is a set of $n + 1$ points and these points define n Euclidean vectors with a specified direction, initial point and length. The length of the HL-Event is a sum of n Euclidean vector lengths. Basing on these assumptions the set of mouse activity characteristics can be designated. The characteristics can be divided into following groups.

The first feature is velocity. The velocity at the first point of the HL-Event is zero $(v_0 = 0)$. The analyzed sequence of following n points allow to calculate the n successive values of local velocities $V = \{v_0, v_1, v_2, \ldots, v_n\}$. For each HL-Event *the maximal velocity*, the *arithmetic mean* and *median* of velocity were determined together with the *average horizontal* and *vertical velocities*. Additional features connected to velocity are: *jerk* - the mouse cursor acceleration at the start of the movement, *braking* - after determining the point where the velocity had a maximal value it is possible to calculate the braking as a ratio of braking distance to the entire distance traveled by the cursor.

Additionally the areas in the close neighborhood of the mouse click event (click neighborhood CN) were analyzed. CN is a square shaped area of side 100 pixels and the center at the mouse click coordinates. The example of activity in CN is presented in (Fig. 2a) - the big dot represents the mouse click event and the small dots coordinates of mouse move events. The left or right mouse button clicks are not distinguished, all basic mouse events in the CN are considered - both before and after the click. First the times were calculated: *waiting time* - the difference between the time when the cursor got to the click coordinates and the click event itself, *hold time* - time between button down and up events - to eliminate high values during DaD actions the value is zero, *reaction time* - time of mouse inactivity at click coordinates - in case of double-click event its counted from second click event. *Correction length* - describes how much too far did the mouse cursor move behind the click coordinates and then came back. It was assumed that the correction length is connected to velocity - higher the velocity of the cursor longer the expected correction length. This relation is described by *correction-velocity ratio* calculated as a ratio of the velocity before the click to the correction length. The last feature in this group is the *approaching* feature describing how does the velocity deceleration look like when approaching to the click coordinates.

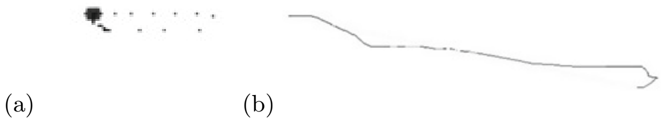
(a) (b)

Fig. 2. Exemplary CN (a) and the trajectory of the HL-Event (b)

The basic feature is the *length* of the trajectory - the real distance traveled by the cursor between the border points. The cursor trajectory between the mouse button click events is an irregular line rarely covering the straight line passing through these points. The cursor trajectory features describe how the trajectory differs from the straight line: *filling* - the area between the trajectory and straight line in pixels, *filling ratio* - calculated by dividing the filling by the length, *filling-distance ratio* - the ratio of filling to the shortest distance between the border points, *deviation from the line* - standard deviation of the distances between the trajectory and the shortest distance, *maximum curvature* - the maximal change of the direction of the cursor.

The 3 features defined in [4] were used: trajectory center of mass *TCM*, scattering coefficient *SC* and velocity curvature *VCrv*. Additionally the *centre of mass coefficient* was defined as a difference between the TCM and the centre of mass of the straight line, together with the two features describing how much the trajectory differs from the straight line: *trajectory correctness* - calculated as ratio of length to the shortest distance between the border points and *trajectory excess* - the difference between the length and the shortest distance divided by the shortest distance between the border points.

5 User Profiling and Verification

In order to classify a user working with the computer system (as legitimate or non legitimate) it is necessary to designate user's profile. For classification method using k-NN classifier a profile should consist of reference samples that characterize the class. Having a profile of activity it is possible to compare data samples with the profile to determine class membership. In case of the presented approach, having the profile of an authorized user, a comparison of currently working user's activity samples with authorized user's reference profile is performed. To create a profile a set of training data consisting of 100 HL-Events of the authorized user is used - the number of HL-Events was determined experimentally. In this study the profile was defined as the vector of the average values of all the features extracted from the training set calculated as the arithmetic mean of the feature values.

5.1 Profile Verification

There are two types of classification: classification by determining the similarity of the sample to the class representative (hereinafter referred to as verification), and classification by defining classes based on the defining properties. The presented approach uses classification method to verify the user so the first type of classification is performed. The verification methods determine the similarity of the samples to previously designated reference profile - a representative of the class. A high rate of similarity proves the consistency of data. With this approach it is easy to analyze the data in order to detect anomalies, which are characterized by a smaller value of the similarity to the profile. The key issue is to establish the threshold determining whether the sample and profile belong to the same class. The result of the method may take the form of binary information: compliance or lack of compliance with the model.

The approach introduced in this paper is based on the k-NN classifier as a verification method. Since the studies relate to the process of verification and not classification, it is necessary to modify the k-NN algorithm - its role was limited to the determination of the similarity between the profile and the subsequent events of the tested set. The k-NN algorithm generally calculates the distance or similarity between the two objects. To determine the similarity between the k features of the tested set (A) and the profile (B) the following distance measures were used: *Euclidean Distance* d_E (1), *City Block* d_{CB} - also called Manhattan Distance (2), *Canberra Distance* d_C - a modified version of City Block (3) and *Bray Curtis Distance* d_{BC} - the result has a normalized form and therefore the result values are in the range of $< 0, 1 >$, where zero indicates that objects are identical (4).

$$d_E(A,B) = \sqrt{\sum_{i=1}^{k}(x_{Ai} - x_{Bi})^2} \tag{1}$$

$$d_{CB}(A,B) = \sum_{i=1}^{k} |x_{Ai} - x_{Bi}| \tag{2}$$

$$d_C(A,B) = \sum_{i=1}^{k} \frac{|x_{Ai} - x_{Bi}|}{x_{Ai} + x_{Bi}} \tag{3}$$

$$d_{BC}(A,B) = \frac{\sum_{i=1}^{k} |x_{Ai} - x_{Bi}|}{\sum_{i=1}^{k}(x_{Ai} + x_{Bi})} \tag{4}$$

5.2 Verification Improvement by Median Filter

Each method of biometric data acquisition is imperfect and is associated with the acquisition of relevant data as well as noise and distortion. The same situation takes place during the process of registering mouse movements. Each type of computer mouse is characterized by a limited sensitivity, which more or less affects the quality of the collected data. External factors such as dust accumulating on the lens of the mouse, or inequality and contamination of the surface on which the mouse is used affect the formation of interference in the data collected. The resulting noise can have a negative impact on the effectiveness of the analysis. Therefore, it is necessary to consider eliminating unwanted noise information, for example by performing the filtering process.

In this paper the use of median filtering is proposed, which is used primarily to reduce the distortion in digital images. For the purposes of performed work the filtering method has been adapted for the use with numbers. The median filter is composed of two operations: the sorting of the data and then extracting the middle values. In this way, the extreme values and outliers are discarded.

6 Experiments and Results

To evaluate the proposed solutions the designation of the EER was chosen. The lower the EER value of the more effective the method. The final EER value for the method is the arithmetic mean of the EER values obtained in all the experiments.

The number of samples was selected experimentally. The study showed EER of 36% for testing set consisted of 100 samples. The EER of the method for 200 samples raised to 42% to descend again to a level of 37% for 1000 samples. It should be noted that the aim is to obtain a method of the on-line type hence the number of samples for a single experiment should be low - for this reason it was decided to use 100 samples in the experiments. For each tested user the profile was created and then the similarity between the profile and a testing set was calculated. Testing set consisted of 200 random samples: the first 100 samples belong to the same user as the profile and further 100 samples to another user constituting the intruder.

6.1 Distances

The k-NN algorithm used in the experiments is based on a distance measure between two object (profile and tested sample). The smaller the calculated distance, the greater the compatibility of the two objects. The next stage of the research was to compare the proposed distance measures. To this end, users were put together in pairs in various combinations. One of the users was an authorized user while the second was an intruder. Then the method using proposed distance measures was tested. The results of the experiments presented in Table 1 indicate that the best method of calculating the distance of the proposed features is the City Block, for which the value of the EER is 30%. The EER value of 30% is still too high for user verification method. Therefore, it was necessary to modify the methods to improve its effectiveness.

Table 1. EER values for examined distance measures

Distance measure	EER
Euclidean Distance	30.88%
City Block	30%
Canberra Distance	36%
Bray Curtis Distance	36%

6.2 Filtering the Distances

When analyzing the distances between the profile and the samples it was observed that these distances are characterized by a high disharmony, which increases the EER. It is necessary to strive for the elimination of extreme values, for example by applying a median filter. Filtration has been designated to the distances and it was a success.

The calculated EER confirmed the achievement of objective pursued. In some of the experiments based on the Euclidean distances the EER was 12% (30% without filtering) and the average EER was decreased by 12.88% and after applying the filtering was 18%. However, the best results were obtained after applying the filter for City Block distances. The lowest value of EER for an experiment was approximately 11% and the average EER was 17.88%.

7 Conclusions

The main goal of the conducted research was to develop the biometric method of user verification based on the analysis of user's activity connected to mouse use. Additionally the method should allow the analysis of continuous activity of a user in an on-line mode. In order to achieve the goals the set of mouse activity features was defined basing on previous works and extended by introducing new features. A new method of aggregating the mouse activity basic events into a higher level event was also presented. Additionally, an attempt to intrusion detection based on the k-NN classifier and various distance metrics was performed. To improve the effectiveness of the method the calculated distances were filtered by using median filter what improved the EER of the method by more than 12%. The best results in both cases (with and without filtering) were observed for the City Block distance measure.

Despite the fact that the article presents the preliminary research end conclusions the results are promising. In future works the further improvements are necessary. The arithmetic mean used to create the profile is very sensitive and a large impact on its value has the noise in the activity data. To reduce the influence of outliers a different model for creating a profile using e.g. the median or dominant could be proposed. Another improvement could be achieved by analyzing the feature set in order to extract the most distinctive features or by using different classifiers.

References

1. Agrawal, A.: User Profiling in GUI Based Windows Systems for Intrusion Detection, Master's Projects, Paper 303 (2013)
2. Ahmed, A.A.E., Traore, I.: A New Biometric Technology Based on Mouse Dynamics. IEEE Transactions on Dependable and Secure Computing 4(3), 165–179 (2007)
3. Banerjee, S.P., Woodard, D.L.: Biometric Authentication and Identification Using Keystroke Dynamics: A survey. J. of Pattern Recognition Research 7, 116–139 (2012)
4. Feher, C., Elovici, Y., Moskovitch, R., Rokach, L., Schclar, A.: User Identity Verification via Mouse Dynamics. Information Sciences 201, 19–36 (2012)
5. Gamboa, H., Fred, A.: A Behavioral Biometric System Based on Human Computer Interaction. Biometric Technology for Human Identification, 381–392 (2004)
6. Garg, A., Rahalkar, R., Upadhyaya, S., Kwiat, K.: Profiling Users in GUI Based Systems for Masquerade Detection. In: Proc. of the 7th IEEE Workshop on Information Assurance, pp. 48–54 (2006)
7. Hsing-Kuo, P., Junaidillah, F., Hong-Yi, L., Kuan-Ta, C.: Trajectory analysis for user verification and recognition. Knowledge-Based Systems 34, 81–90 (2012)
8. Kaminsky, R., Enev, M., Andersen, E.: Identifying Game Players with Mouse Biometrics, University of Washington, Technical Report (2008)
9. Palys, M., Doroz, R., Porwik, P.: On-line signature recognition based on an analysis of dynamic feature. In: IEEE Int. Conf. on Biometrics and Kansei Engineering, pp. 103–107. Metropolitan University Akihabara, Tokyo (2013)
10. Porwik, P., Sosnowski, M., Wesolowski, T., Wrobel, K.: A computational assessment of a blood vessel's compliance: A procedure based on computed tomography coronary angiography. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS, vol. 6678, pp. 428–435. Springer, Heidelberg (2011)
11. Pusara, M., Brodley, C.E.: User re-authentication via mouse movements. In: Proc. of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, pp. 1–8 (2004)
12. Raiyn, J.: A survey of cyber attack detection strategies. International Journal of Security and Its Applications 8(1), 247–256 (2014)
13. Salem, M.B., Hershkop, S., Stolfo, S.J.: A survey of insider attack detection research. Advances in Information Security, vol. 39, pp. 69–90. Springer, US (2008)
14. Schonlau, M.: Masquerading user data, `http://www.schonlau.net`
15. Wesolowski, T., Kudlacik, P.: Data clustering for the block profile method of intruder detection. J. of MIT 22, 209–216 (2013)
16. Wesolowski, T., Kudlacik, P.: User profiling based on multiple aspects of activity in a computer system. J. of MIT 23, 121–129 (2014)
17. Wesolowski, T., Wrobel, K.: A computational assessment of a blood vessel's roughness. In: Burduk, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnierek, A. (eds.) CORES 2013. AISC, vol. 226, pp. 231–240. Springer, Heidelberg (2013)