Behavioral Mouse-Movement Analysis for User Authentication

Yeganeh Azmoudeh

University of Zanjan

tobe.ywmz@gmail.com

ABSTRACT

OS Hardening and security are such important issues which have been discussed over years and through many different ways. One good way can be “Mouse Movement” tracking and detecting the behavior. Every user has their own way of moving the mouse which can help us identify them within other users, and since most of the times other users using the system can be a threat to our system security in many cases it can be a good way to harden our OS. This method can help us either identify a user or tell if a specific user is using the mouse or not. This can be traced or detected with different methods, but in this particular paper we are going to discuss it through a set of parameters which we will talk about each as well later in this paper. This paper also goes through two main phases for the so called purpose which are: 1) Mouse Movement Tracker. 2) Mouse Movement and User Specification.

INTRODUCTION

As we discussed this method is used to specify an identical user using the mouse. This can be done by some specific parameters through 2 main phases. The parameters used for this purpose are: Speed, Click Frequency, Path Curvature, Dwell Time, Idle Time. There are more parameters which can be used in this project but we use the ones we mentioned. We will discuss about each parameter and how they can be effective for our purpose in this paper.

OVERVIEW

So far we know that we need to use some parameters, these parameters have to be calculated based on the collected data in the first phase which is “Mouse Movement Tracker”. In this phase we collect some data that specifies the user’s behavior. Collecting the data, we need to capture mouse events. In this project we capture three kinds of events: 1) Move: We use the “on_move” event to trace the path with getting x and y which locates the mouse. 2)Click: We use the “on_click” event to whether our user is clicking on something or not. 3)Scroll: We use the “on_scroll” event to specify whether our user is scrolling or not.

We also have a “Time Stamp” which collects the exact time of these events. This data will be saved in a log file. These three events will help us calculate our parameters:

1.Speed: To calculate the speed threshold, first we calculate the mouse speed for two movement events in a row. To do that we calculate the distance between two points divided by the time difference calculated from the log’s timestamp. We repeat this process for each move event pairs and we would store the calculated speed in an array. Finally, we would calculate the mean of the stored speed. This value would be the dynamic threshold for the current user.

2.Click Frequency: To calculate click frequency, first we would find all click events (in this stage we are only interested in the pressed clicking events) then we would calculate the time difference between each click event and store the results in a separate array. After processing all click events, we would calculate the mean of the time differences. This value would be the dynamic threshold for the current user.

3.Path Curvature: To calculate path curvature, we would find three move events in a row. The goal is to calculate the angle that these three move events make. To do so we create 2 vectors (point1 to point2, point2 to point3). We store these angles in a separate array to calculate the mean value as the path curvature threshold for the current user.

4.Idle Time: We would consider a time period of half a second with no events an “idle time”. This basically indicates the time periods when mouse is not moved, clicked or scrolled. To calculate a threshold for this parameter, we would look through all locked events regardless of their type and calculate the time difference between each event pair. If the time difference is greater than a specific value (in our case, half a second) it is considered an idle time and stored in an array. After repeating the same process for all of the captured events we would calculate the mean value as we did in previous steps.

5.Dwell Time: “Dwell time” in the context of mouse movement refers to the amount of time the mouse pointer remains still. It could indicate user attention or user making a decision. To find dwell times, we look for move events in which the “x” and “y” values stay the same. Then we would calculate the time difference between these events, and the mean of these values would be dwell time threshold for the current user.

The measured parameters are calculated after a baseline data-collection phase (as we referred to as phase1) is completed, the time period of this phase could vary. We started our tests with a baseline data-collection time period of ten seconds. It turned out that in a short time period, not enough mouse events could be captured to calculate precise parameters for a specific user, since the program would detect a user change even when the user was still the same.

In order to increase the accuracy of the parameters we increased the time period in which a baseline-data would be collected. After testing with different time periods, we came to conclusions that the time period of fifty seconds would be reasonable to collect baseline data, since it was neither too short nor too long and could detect the user change correctly.

Since the user behavior is not exactly the same at all times, a mouse behavior change is expected although the user is still the same. To adapt to user behavior change we add a tolerance such that a sixty percent match of user behavior would be enough to indicate that the user is the same.

Conclusion

We have presented the “Behavioral Mouse-Movement Analysis for User Authentication” which analysis user behavior to detect user change. This project was a simple demo of what would could use user behavior for security purposes. This feature could be extended in many different ways such as sending alerts to a user when user change is detected or locking the system when too many user changes are detected. Our project could be used for highly secured systems where user change is a serious concern. This project could also be deployed on systems with high-access when a system auto-reaction could be life saving.