Authentication Enhancement by Keystroke Dynamics: Another Dimension of the Password

Myungho Jung
University of Utah
myungho.jung@utah.edu

I. INTRODUCTION

<Motivation>

Password authentication has been one of the most popular method to identify. It is ubiquitous for systems such as Web sites, SSH, and mobile phones. However, attackers' techniques to steal password have also been developed in many ways. Therefore, we need to collect additional information to figure out identity other than just plain text of passwords. For example, access from different IP or MAC address can be suspected as an attack. However, it makes users inconvenient if they are travelling or buy a new device. Keystroke dynamics can be extracted while typing password as well as used as identity verification[3]. And there exists many commercial cases[1].

In this project we focused on how to implement and optimize the mechanism in practice.

II. RELATED WORKS

Many algorithms to detect outliers are studied and test on performance and precision[**?**]. Some of algorithms show very high accuracy in detection; however, the computation is not fast enough to apply to web services which has millions of users.

III. METHOD

A. Data structure

The timing vector of keystroke patterns can be measured as intervals between keys and keystroke durations[2]. Therefore, a pattern of $n+1$ numbers will be created from typing n characters password including return key. In most studies, the vector is measure like this; however, we applied it in a slightly different way. The interval in our project measured not from previous release time but from previous press time. By doing so, there is no negative value and storage can be saved by store only unsigned data.

The timing vector can be considered as a point in $n+1$ dimensional space. Then, we can measure the distance of two points, and apply machine learning or neuron network algorithm to detect novelties.

B. Normalization

Pattern data should be normalized for storage and precision. Assume that the timing are stored in nanoseconds. Then, more bytes should be allocated to store slow typing users' patterns than faster users. On top of that, if the patterns are stored in milliseconds, the short timings in the pattern will result in less accurate detection. Therefore, the data should be normalized based on the last timing element, and also be normalized for equivalent comparison.

1. Normalization for storage

2. Normalization for precision

C. Application scenario

This mechanism can be applied for actual services by two ways. First, it can be used as an authentication. For example, if the user's input pattern is different from the original pattern, the user's access is rejected. On the other hand, we can think that it is just an additional process to enhance security. In this case, the system will notify the user of warning message by email or text even if the input pattern is detected as an outlier.

Error rate for pattern detection can be divided into two parts which are False Acceptance Rate(FAR) and False Rejection Rate(FRR)[2]. Which rate is important in which cases? FAR is the most important in the case that keystroke dynamics is directly applied for authentication. As many attachers' patterns as possible should be rejected although some of the actual user's pattern is not verified. If the algorithm is secondary method for security enhancement, FAR is less important than the first case. Moreover, many warning messages caused by high FRR will make the users inconvenient.

IV. ALGORITHM

A. Novelty Detection

To filter imposter's patterns, a novelty detection algorithm is required. There are many researches to detect anomaly. The algorithms can be divided into statistical type, neural network, pattern recognition based on learn-

ing, and heuristics[1]. We implemented and tested four common algorithms from them.

1. Statistical Algorithm

Algorithms using distance, mean, and standard deviation are included in this section. For the project, three algorithms are implemented using Euclidean distance, Manhattan distance, mean and standard deviation.

2. Machine Learning

To compute more sophisticated results, we applied a machine learning algorithm which is one-class Support Vector Machines(SVMs). One disadvantage of this method is requiring many patterns in advance. Regardless of the defect, the more we collect data, the more accurate result we can expect.

V. TEST

The performance of statistical and machine learning algorithms have already been tested in many studies. Thus, we focused on expecting actual users' patterns. Previous works are based on heuristic results or similar dataset. However, the pattern may not be even even if the same person types several times. It would be came from using different type of input devices or physical changes. For instance, it will show different patterns when using PC and keyboard than using mobile devices. Also, the more familiar users become to type the same password, the faster they type.

A. Environment

1. Statistics

Unlike other researches, we simulated changing patterns of a user. There are two two possible changes in a user's patterns. First, only an interval of keystroke rhythm would be different each time. Suppose that the password is mixed with a sequence of alphabets and that of numbers like 'abcd1234.' In this cases, the user's patterns between word and numbers may not be expected. Nevertheless, we can distinguish patterns because most of entities of the imposter's pattern will far different from the original pattern although there would be only one or two different intervals in the genuine user's patterns. In addition, the user's pattern will change only proportionally. It would come from that the user become familiar with typing password or just be able to type faster than before. Moreover, the user may use different types of keyboards such as portable keyboard or touch keyboard on mobile devices other than regular keyboard.

VI. IMPLEMENTATION

VII. DEFECTS

Although there are many researches and practical application of Keyboard Dynamics, there are problems in authenticating with the method. Assume that the mechanism is applied for Web-based system. Then, people can move cursors using arrow keys or mouse while typing and editing password. We just ignored the cases and assumed that the web page allows only password typed at once. Though it would make the users inconvenient, the web service will become more secure. If the mechanism is used for SSH, we don't have to worry about problems caused by arrow keys and mouse. However, there is still problems on pressing backspace key. For security enhancement, it is necessary to block the key inputs and use of mouse on the password window.

A. Adversary Models

VIII. CONCLUSION

[1] Salil P Banerjee and Damon L Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012.

[2] Sungzoon Cho, Chigeun Han, Dae Hee Han, and Hyung-Il Kim. Web-based keystroke dynamics identity verification using neural network. *Journal of organizational computing and electronic commerce*, 10(4):295–307, 2000.

[3] R Stockton Gaines, William Lisowski, S James Press, and Norman Shapiro. Authentication by keystroke timing: Some preliminary results. Technical report, DTIC Document, 1980.