Brief Announcement: Deriving Context for Touch Events*

Moran Azran, Niv Ben Shabat, Tal Shkolnik and Yossi Oren

Department of Software and Information Systems Engineering, Ben Gurion University, Beer Sheva

Email: {azranmo@post.|nivb@post.|talshko@post.|yos@}bgu.ac.il

Abstract. To quantify the amount of high-level context information which can be derived by observing only a user's touchscreen interactions, we performed a user study, in which we recorded 160 touch interaction sessions from users running different applications, and then applied both classical machine learning methods and deep learning methods to the results. Our results show that it is possible to derive higher-level user context information based on touch events alone, validating the efficacy of touch injection attacks.

Keywords: machine learning, malicious hardware, smart phone.

1 Introduction

Smart phone touchscreens are often produced by third-party manufacturers and not by the phone vendors themselves. According to a 2015 study, more than 50% of global smartphone owners have damaged their phone screen at least once, and 21% of global smartphone owners are currently using a phone with a cracked or shattered screen [1]. These shattered screens are often replaced with aftermarket components of questionable origin. In [2,3], Shwartz et al. showed how malicious touchscreen hardware can launch a **touch injection attack** that allows the touchscreen to impersonate the user and exfiltrate data. One limitation of this attack approach is that the attacker knows the position and timing of touches on the victim's screen, but does not have any higher-level **contextual** information such as the user's current activity or current running application.

The main objective of our research is to quantify the amount of high-level context information the attacker can derive by observing only the user's touchscreen interactions. If an attacker can understand the context of certain events, he can use this information to create a customized attack which will be more effective. For example, the attacker can know when he should steal information from the user or to insert malicious touches. To quantify the amount of high-level context information which can be derived by observing only a user's touchscreen interactions, we performed a user study, in which we recorded 160 touch interaction sessions from users running different applications.

2 Method and Results

2.1 Experiment Setup

The experiment was conducted on a group of third year university students. In the first part of the experiment, the subjects were required to fill in a personal survey which included questions

* This research was supported by Israel Science Foundation grants 702/16 and 703/16.

such as: age, gender, which hand do you usually hold the cell phone? Do you usually hold the cell phone with both hands? when did you last play on your cell phone? When was the last time you drank coffee? In addition, subjects were asked whether they knew certain games. In the second stage of the experiment, each subject was asked to record four different touch interaction sessions on the test phone. First, the subjects were asked to play the game "Color Infinity". The objective of this game is to pass the ball through various obstacles. The game included fast and short touches around the screen. Next, subjects were required to play a game called "Bricks". The objective of this game is to move the bricks to the appropriate color when at some point the color of the brick changes. This game includes continuous screen touches. When the subjects finished playing games, the subjects were asked to launch the phone's web browser and perform a web search by typing the word "Facebook" in the browser search bar. Finally, the subjects were asked to enter an e-mail application on the cell phone and send an e-mail containing a subject and content line.

2.2 Machine Learning Methods

Feature selection: The features we selected for classical machine learning were derived from the features described in [4], including median velocity of the five last points of the trajectory, mean resultant length, largest absolute perpendicular distance between the end-to-end connection, stroke duration and inter stroke time. We augmented the feature set of [4] with three additional features suggested by Meng et al. in [5]: average touch movement speed per direction, average single-touch/multi-touch time and number of touch movements per stroke (NTM).

Classifier selection: We evaluated multiple ML models, including Logistic Regression, Linear Discriminant Analysis, K Nearest Neighbors, Decision Tree, Gaussian Naive Bayes, Random Forest and Quadratic Discriminant Analysis. All models were instantiated using their default parameters. To compare the performance of classical ML algorithms with deep learning algorithms, we also analyzed the raw touch information using a deep learning convolutional neural network (CNN). Our CNN had three Conv1D layers, three MaxPool1D layers and a final softmax activation layer.

2.3 Data Collection and Initial Processing

Data collection was conducted on 01/03/2018 during a university hackathon event. We collected 153 touch recordings from 40 different subjects. The experiment took 4 hours in total. To record the touches, we used a specially modified LG Nexus 5X Android phone. The phone was modified at the root-kit level with a touch recording functionality, which runs in the background and outputs a CSV file with the touch screen locations, pressure and timestamp. Data from the phone was downloaded to a workstation running Matlab and Python for further analysis. For classical machine learning we used Matlab's Classification Learner tool any Python's scikit-learn toolkit. For deep learning we used the TensorFlow framework running on Python.

2.4 Machine Learning Results

The performance of the classical machine learning classifiers is summarized in Table 1 .The classical machine learning classifiers were highly effective in determining the activity context of the user from the supplied touch data, with the best-performing classifier (Linear Discriminant Analysis) providing a prediction rate of over 92% over the 4 activity contexts evaluated. The relative ranking of the different features, as output by the relieff algorithm, is summarized in Table 2, and shows

Algorithm	Prediction Rate
Logistic Regression	0.8954
Linear Discriminant Analysis	0.9215
KNeighbors	0.8039
Decision Tree	0.8692
GaussianNB	0.9281
Random Forest	0.9019
Quadratic Discriminant Analysis	0.8954

Table 1. Performance of Classical Machine Learning Classifiers

Predictor Rank	Feature	Predictor Importance Weight
1	Stop_Y	0.1938
2	Y_Avg	0.1913
3	Stop_X	0.1790
4	Stroke_Duration	0.1275
5	Start_X	0.0978
6	Start_Y	0.0895
7	Pressure_Avg	0.0634
8	X_Avg	0.0575

Table 2. Predictor Ranks for Context Recognition (as output by the relieff algorithm)

that the most significant features are the final Y coordinate and the average Y coordinate of each stroke.

We ran our deep learning classifier on the raw data with 30 epochs and a 90-10 validation_split. The deep learning classifier was able to detect the correct activity 87.5% of the time on the validation set, a level of performance similar to that of the classical methods.

3 Conclusion

Our results show that it is possible to derive higher-level user context information based on touch events alone, validating the efficacy of touch injection attacks. Applying touch context analysis on the defensive side can also have a benefit, since it can prevent attacks by identifying anomalous interaction and therefore protect against abnormal use of the phone.

References

1. Motorola Mobility. Cracked screens and broken hearts - the 2015 motorola global shattered screen survey. `https://community.motorola.com/blog/cracked-screens-and-broken-hearts`.
2. Omer Shwartz, Guy Shitrit, Asaf Shabtai, and Yossi Oren. From smashed screens to smashed stacks: Attacking mobile phones using malicious aftermarket parts. In *2017 IEEE European Symposium on Security and Privacy Workshops, EuroS&P Workshops 2017, Paris, France, April 26-28, 2017*, pages 94–98. IEEE, 2017.
3. Omer Shwartz, Amir Cohen, Asaf Shabtai, and Yossi Oren. Shattered trust: When replacement smartphone components attack. In William Enck and Collin Mulliner, editors, *11th USENIX Workshop on Offensive Technologies, WOOT 2017, Vancouver, BC, Canada, August 14-15, 2017*. USENIX Association, 2017.
4. Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Trans. Information Forensics and Security*, 8(1):136–148, 2013.
5. Yuxin Meng, Duncan S. Wong, Roman Schlegel, and Lam-for Kwok. Touch gestures based biometric authentication scheme for touchscreen mobile phones. In Miroslaw Kutylowski and Moti Yung, editors, *Information Security and Cryptology - 8th International Conference, Inscrypt 2012, Beijing, China, November 28-30, 2012, Revised Selected Papers*, volume 7763 of *Lecture Notes in Computer Science*, pages 331–350. Springer, 2012.