Improving Android's Device Security using Behavioral Biometrics

Yonatan Mekonnen
HiLCoE, Software Engineering Programme, Ethiopia
yoniecco@gmail.com

Gezahegne Dugassa
HiLCoE, Ethiopia
gdugassa@gmail.com

Abstract

Mobile phone technology dominantly controls our daily life whether for solemn business matter or for our social life through which we access the whole universe just in a tap away. Thus, the power of connectivity using the Internet through GSM (Global System for Mobile)/LTE (Latest Technology Extended) enhanced its usage in a way not having one can be considered as handicapped. Losing our device, leaving the device unattended and exposure of our password/pattern can make us not only lose our reputation in our social network presence but also might be a risk allowing others to have control over our activities by the features we use through applications in our phones.

Hence, authentication becomes one of the core issues in security as one is privileged to do any activities and tasks so long as s/he provides the credentials and is authenticated regardless of the real identity. Relying on the default screen-lock option that Android devices have is common. The problem here is that this screen-lock authentication feature is breakable and the device can be unlocked easily.

Thus, the proposed solution in this paper is designed to act as a layer for the device's security improving confidentiality of its content when the screen lock feature is compromised due to various reasons. Our application will manage to work on behavioral authentication by studying and analyzing the characteristics of the owner of the apparatus with a combination of signature and character usage scheme. This paper adopted the concept of machine learning from natural language processing so that an intelligent application can secure the device's content from unauthorized access.

Keywords: Authentication; Security; Mobile Apparatus; Screen Lock

1. Introduction

1.1 Background

Mobile devices usually contain information like the contact list which were intended to be kept by ourselves up to confirmation and verification messages that will enable us change passwords for our official/personal emails, mobile banking applications and social media accounts. Though we use the applied "Screen-Lock" option by Android, bypassing it is actually as easy as plugging the locked device into a computer that runs applications which can remove the Screen-Lock in just a click [1].

The irony here is that this Screen-Lock removing software are available in different options and feature containers with options like simple designs to a very complex one for free. Therefore, it is no more a wonder to see one's device being "Unlocked". Android actually has shown improvement in their versions for security but it is not as strong as the potential that exist working against it. Android is open-source and the Software Development Kit (SDK) it uses allows third-party applications and developers to gain access deep into the Unix-kernel through rooting the device or even by just asking for permission upon installation [2].

As stated in [1], because of Android's powerful development framework, users as well as software developers are able to create their own applications for wide range of devices. Some of the key features of Android operating system are: Application Framework, Dalvik virtual machine, Integrated browser, Optimized Graphics, SQLite (Server Query Language), Media Support, GSM Technology, Bluetooth, Edge (Enhanced Data for GSM

Evolution), 3G (third Generation), WiFi, Camera, and GPS (Global Positioning System). To help the developers for better software development, Android provides SDK.

The intention we have here is as to how can we improve the device security when the authentication that Android implemented gets breached by designing an application to act as another layer of security so that we at least can maintain Confidentiality.

The following research questions are formulated:

1. What is Android's Screen-Lock authenticator?
2. How does Android's Screen-Lock authentication work?
3. How strong is Android's Screen-Lock authenticator and what are its limitations?
4. How can we improve the limitations?

This paper puts a focus mainly on understanding and analyzing the gap and limitation that the default Android Screen-Lock authentication has. The solution that is proposed will be applied only for devices that run Android operating system which allows integration of third-party applications.

The proposed solution in this paper will equip the end user with the following.

- A cost-effective and user-friendly security improvement mechanism.
- Secure the device so that it will be safer to save data and use applications.
- Avoid the requirement of additional hardware for authentication.
- Save the hassle of fighting to secure the content of the device.

1.2 Methodology

Experimental research method is recommended by most scholars as it is highly effective on problems that require complex software solutions such as the creation of software development environments, the organization of data that is not tabular, or the construction of tools to solve constrained optimization problems.

While applying it to our specific problem area, the method was used to extract flaws of Android's screen-lock authenticator and assess the possibility of improvement. Also, the method was used to test the sample taken with the algorithms proposed and choose one from the three. As this approach is largely used to identify concepts that facilitate solutions to a problem and then evaluate the solutions through the construction of prototype systems, it was found convenient and applicable with better throughput than other research methods.

Thus, we used qualitative research method enabling us to fulfill our objectives, use philosophical assumptions for understanding how people make sense of their worlds and the experiences people have along with knowing or understanding from the participants' perspectives on what is needed to be addressed. It also has enabled us to address what is needed to be given a key focus for understanding, rather than predicting or controlling, social settings or social phenomena while intending to address the problem specified.

Opportunity sampling was used to select the test subjects as this sampling method uses people from the target population available at the time and willing to take part. However, for the evaluation of the experiment, we used stratified sampling as we identified the different types of people that make up the target population and works out the proportions needed for the sample to be representative. Since gathering such a sample would be extremely time consuming and difficult to do, we recreated the samples by generating conditions using a simple randomization algorithm.

We added random sampling to strengthen the evaluation using Weka and MS Excel along with simple randomization algorithm so that we can measure cost, efficiency, accuracy and more detailed performance related aspects. The advantage is that our sample will represent the target population and eliminate sampling bias caused by opportunity sampling even though it requires more time and effort.

As natural language processing is applied for our application, we gave emphasis for the following 6 key attributes in which our decision will be determined by.

1. TP = true positives: number of examples predicted positive that are actually positive
2. FP = false positives: number of examples predicted positive that are actually negative
3. TN = true negatives: number of examples predicted negative that are actually negative
4. FN = false negatives: number of examples predicted negative that are actually positive.
5. Accuracy: the degree to which the result of a measurement, calculation, or specification conforms to the correct value or a standard.
6. Precision: refinement in a measurement, calculation, or specification, especially as represented by the number of digits given.

We then examined and got a data of 31 records for each of the 4 conditions in our equation specified as Signature (S), Typing Behavior (T), Favorite A/V (A), and Frequently Accessed apps (F). We then populated a data of 6000 for each condition with total evaluation data of 6031. Out of the 6031, we only fed 40 true (valid) conditions using a logical operator AND to our CSV file where the A cells contain Signature, the B is for Typing Behavior, C contains Favorite Audio/Video files and D is for Frequently accessed applications as follows: AND(A2>=94, A2<=100, B2>=85, B2<=100, C2>=80, C2<=100, D2>=90, D2<=100).

2. Related Work

Application markets such as Apple's App Store and Google's Android Market provide point and click access to hundreds of thousands of paid and free applications. Markets streamline software marketing, installation, and update therein creating low barriers to bring applications to market, and even lower barriers for users to obtain and use them. The fluidity of the markets also presents enormous security challenges. Rapidly developed and deployed applications [3], coarse permission systems [4], privacy invading behaviors [5, 6, 7], malware [8, 9, 10], and limited security models have led to exploitable phones and applications. Although users seemingly desire it, markets are not in a position to provide security in more than a superficial way [11]. The lack of a common definition for security and the volume of applications ensure that some malicious, questionable, and vulnerable applications will find their way to the market [9].

Biometric authentication technologies are used for machine identification of individuals. The human-generated patterns used may be primarily physiological or behavioral, but usually contain elements of both components. Examples include voice, handwriting, face, eye and fingerprint identification. The search is to find a uniquely identifying characteristic not of what we are, but of what we do. An example is gait analyze someone's walking style and one can easily determine their identity. However, that is not going to work on a Smartphone. But if we consider a person's signature that is the only security layer in banking. It can be analyzed since exact handwriting style is unique to everyone. It is possible that devices could analyze the speed, style and exact position on the screen of how one signs a name, probably using a stylus [12].

There are two types of signature features that are used for signature verification: Functions and Parameters.

1. Function features are used in terms of time functions whose values constitute the feature set. Parameter features are used in terms of a vector of elements, each one representative of the value of a feature. Function features allow better performance than parameters but require higher system cost for matching [13].
2. Parameter features are classified into two categories: Global and Local. Global parameters concern features that are related to whole signature, i.e., total writing time, number of pen strokes in the signature and total written distance. Local features are like

global features but retrieved only from the specific part.

Depending on the level of details, local parameters can be divided into two categories: component-oriented and point (pixel) oriented.

1. Component oriented parameters are extracted at the level of component, i.e., stroke height, width ratio and relative position of signature parts.
2. Point oriented are extracted at levels of point (pixel) resolution, i.e., pixel density and the total count of the stylus points. This feature classification is not strict. Some features can be used globally and locally.

The following are some of the most common function and parameter features found in the literature [14].

- Position: X and Y position of pen tip
- Speed: Overall speed of writing or in X and Y direction
- Acceleration: Overall acceleration or in X and Y direction
- Pressure: pressure on which pen leaves to the signing area
- Direction of pen movement: direction of pen speed vector

Our application will use offline verification though the attributes position, speed and acceleration are widely used for online signature verification. But for our application, velocity and acceleration function can be obtained from a specific device or numerically derived from position. Pen inclination is considered as a most consistent feature. However, it can be captured only by some specific devices. Finally, we will take a percentage for the match after comparison is made and will use it for our later equation as a variable S.

Text Prediction method of authentication model will help us in design level as our proposed solution will learn the behavior of one's way of word usage in such a way that a relation can be made between words. After a sensible set of data is developed in the application database, the application will generate or re-use a phrase as a question to be asked during authentication as it will determine the most likely response. Behavioral biometrics will then be applied by calculating how close the similarity is with some tolerable deviation in a percentage scale.

In its most basic form, keyboard prediction uses text that one enters over time to build a custom, local "dictionary" of words and phrases that are typed repeatedly. It then "scores" those words by the probability one uses or needs it again. The first or second time one ignores the word, it will assume it is not a misspelling, but not a word one uses often enough to be presented within similar usage patterns. If one ignores it a third or fourth time (how many times depends on the specific keyboard), the keyboard will mark it as a future probable choice, and start presenting with it when one types similar words or sentences [14].

3. The Proposed Solution

For a developer of Android applications, the SDK will work in a standard and convenient way so long as application permission wizard is designed and configured properly for the SDK to comply with it. As a developer, we are able to design an application which replaces any existing system applications including the pre-installed application like SMS client application, default-dialler, the web browser and contact manager.

The favour we have benefited from Android is that there is no such thing as a compulsory application. Also there is equality among applications and all applications can run in parallel as per their pre-defined priority. Since Android operating system is open source and allows various permissions, an Android application developer can:

1. Integrate a new application along with an existing one.
2. Extend an application by adding some new features in the new one.
3. Replace an existing application with a new one but with similar functionalities.

The good thing about Android's operating system is that a third-party application can be designed in such a way that it can take a full control over the device. This is to mean that if a given application is designed with some specific permission settings, i.e., manifestation for the SDK, then that application will be considered as a trusted application and can run in the same place that the operating system executes.

Therefore, we have designed an application that can act as a layer between the applications and the user. This layer contains the following functions.

1. Hides confirmation messages sent to the device by scanning keywords and phrases that are contained as main structures of confirmation messages.
2. Hides applications that are sensitive and used for authentication like Google's one-time password generator.
3. Initiates device activities remotely including encryption, shred-data, factory-reset and wipe data through USSD (Unstructured Supplementary Service Data) and SMS commands.
4. Sends SMS containing new SIM (Subscriber Identity Module) status to the actual owner of the device once another SIM is inserted into the device by capturing the new SIM-Card's ICCD (Intensified Charge-Coupled Device) and service number.
5. Forces log out any accounts that are logged in when the device detects security breach.
6. Clears cache memories for applications by having control over the application manager module.
7. Re-authenticates the user after a breach is detected using behavioral authentication.

4. Discussion

So far, we have investigated and analyzed the core and basics of Android operating system. By doing so, we are able to assure that the problem does actually exist and many users are being affected by it. For instance, if we take one-day statistical data from ethio telecom and Facebook joint monitoring dashboard portal, approximately 30% of all Facebook users use Android phones on average per day. This is huge number as the "all" refers to those who use the desktop web browser and other mobile operating systems like iOS (iPhone Operating System). This also implies that at the minimum 1000 people per day that ethio telecom gives a SIM card replacement, the same percentage are Android based devices. As our motive mainly relies on a scenario in which a user's device falls in the wrong hand regardless of the cause, the Android based devices will be vulnerable if intended for easy unlock of the Screen-Lock Authenticator.

The proposed application is implemented having the following components for authentication as our application will have 4 authentication attributes:

1. The hand-written signature noted as S
2. Typing behavior noted as T
3. Favorite audio/video file noted as A
4. Frequently accessed application noted as F

While using our application, the user will be asked to select authentication preferences from the above four in a check-list. The user will be forced to select a minimum of two authenticators from the four provided that signature will be considered as a prerequisite since it is the core attribute for our application. Then from the chosen ones, the application will create an arithmetic percentage and then will base its decision on the deviation from the acceptable result. Thus, our authentication equation noted as A will be: A = (R ≤ (%M+(%O)))/O+1 and is mathematically represented as.

$$A = R \leq \frac{(\%M + (\%O))}{O + 1}$$

where:

- A is our application authentication as whole,
- R is Threshold in which is validated to be authenticated or not,

- M is a constant mandatory attribute chosen for authentication, and
- O is an optional variable attribute or attribute chosen for authentication.

Then, deviation will be calculated by defining a range from which a minimum and a maximum can be registered. For instance, in the case of signature verification, the 5 pre-signed signatures will be taken as a benchmark and then any signature signed after that will be compared with those five and the difference and variance it has towards the others will be calculated in percent. Then if the difference is greater than 10%, then the newly signed signature will be rejected and the authentication will be considered as a failure. We have defined a range after we apply experiments using sample subjects to study their behavior and signature. The range can then be used for decision in which a result can deviate from the 100% that we have derived for the 4 conditions earlier as per Table 1.

Table 1: Range Definition

Attribute	*Min*	*Max*	*Deviation*
Signature (S)	94	100	6
Typing Behavior (T)	85	100	15
Favorite A/V (A)	80	100	20
Frequently Accessed apps (F)	90	100	10

By doing so, we can verify if the person being authenticated is actually the real user or not. For this to be applied, we can implement it using the code fragment shown below for a scenario in which the user has chosen all the 3 optional authentication parameters where all the "X" variables represent the minimum threshold deviation and all "Y" variables represent the maximum threshold deviation.

Finally, the rule X = ((S=TRUE) AND (T=TRUE) AND (A=TRUE) AND (F=TRUE)) will be applied for our proposed application as a validation control for decision.

After we have the structure above, we implement it using the following algorithm that can be used in any programming language.

```
function Authenticate($Signature, $Typing_Behaviour, $Favourite_AV, $Frequently_Accessed)
   {
       if($Signature && $Typing_Behaviour && $Favourite_AV && $Frequently_Accessed)
       {
           if(($Signature >= X) && (($Signature <= Y))
           {
               if(($Typing_Behaviour >= X) && (($Typing_Behaviour <= Y))
               {
                   if($Favourite_AV >= X) && (($Favourite_AV <= Y))
                           {
    if($Frequently_Accessed >= X) && (($Frequently_Accessed <= Y))
    {
                           return TRUE;
                       }
                   }
               }
           }
       }
   }
```

Out of 8 available algorithms, three were chosen as they fit most for this experimental research. We first tested the data for evaluation using the classifier rules-part. Rule-based machine learning (RBML) is a term in computer science intended to encompass any machine learning method that identifies, learns, or evolves 'rules' to store, manipulate or apply [6, 7, 8]. As our proposed application derived a rule for decision, it implies that the rules typically take the form of an {IF: THEN} expression, (e.g., {IF 'condition' THEN 'result'}, or as a more specific example, {IF 'Signature matched' AND 'favorite application chosen right' THEN 'grant access'}). We then summarize the important elements in Table 2 so

that we can make a decision by which algorithm we should be working with.

Table 2: Decision Table for Algorithm Selection

Algorithm (Classifier)	*True Positive Rate*	*False Positive Rate*	*Precision*	*Recall*	*Confusion Matrix*	*Response Time (in Seconds)*
Rules PART	0.999	0.05	0.999	0.999	Apr-31	0.11
J48 graft	1	0	1	1	0/6031	0.08
Naïve Bayes	0.997	0.422	0.997	0.997	18/6031	0.03

As a result, we have chosen J48 graft classifier as it granted us with 100% precision, true positive, and recall with 0% false positive assuring us effectiveness. It has slower response time compared to Naïve Bayes, but a slower yet accurate response is better than a faster but compromised one.

5. Conclusions and Future Work

This paper designed a solution as an additional layer for device security for Android running devices to improve the mobile device's security to maintain privacy and content confidentiality. By applying behavioral authentication options, including signature, one's typing behavior and the individual's interest for accessing applications and media, the content residing in the device can get additional security. The proposed application for device authentication will not only secure the privacy of the device content but also will allow the end user to feel more secured. Though this paper cannot grant option to re-gain access to a lost device, it can minimize the hassle the user has to go through to re-gain access to an account if a third party has control over the lost device, i.e., a stolen/lost mobile apparatus.

The proposed solution is a security layer by which addition authentication will be applied. But still, the effectiveness of this application will highly depend on the user's willingness to use the application, the knowledge of the user on basic application usage and the combination of authentication options that the user chooses from the application. If a case study focusing on awareness creation is done, end users can manage the level of vulnerability to which they are exposed.

Future work will mainly focus on integrating the proposed solution with artificial intelligence and neural-biology along with face recognition algorithms so that a device independent yet effective application can be developed.

This solution can borrow and adhere logics from psychology and neural biology so that the level of intelligence can be enhanced by advancing its decision-making schema. In addition, the efficiency of the chosen algorithm might vary based on the data input, thus a better experiment involving different test subjects will be more accurate.

References

[1] "Android Application Security with OWASP Mobile Top 10 2014", Luleå University of Technology, 2014.

[2] William Enck, "A Study of Android Application Security and Internet Infrastructure Security", Pennsylvania State University.

[3] University of Minnesota, "Classification Methods", http://www.d.umn.edu/~padhy005/Chapter5.html, 2015.

[4] E. George, "On-line Handwritten Signature Verification Using Wavelets nd Back Propagation Neural Networks," Sixth International Conference on Document Analysis and Recognition, 2001.

[5] R. Abbas, "A Prototype System for Offline Signature Verification Using Multilayered Feed Forward Neural Networks", Michigan, 1996.

[6] https://source.android.com/security/, Last. Accessed on 13/06/2017.

[7] C. Zhong-Hua Quan, "Hybrid HMM/ANN Based Approach for Online Signature Verification," International Joint Conference on Neural Networks, IJCNN 2007.

[8] L. Z. Xia, "Efficient Object Recognition Using Boundary Representation and Wavelet Neural Network," 4th International Conference On ASIC, 2001.

[9] M. G. Kresimir Delac, "A Survey of Biometric Recognition Methods," 46th International Symposium Electronics in Marine, ELMAR-2004, 16-18 June 2004.

[10] Z. S. Zhang, "Association Rule Mining: Models and Algorithms," 2002.

[11] R. N. Totty, "Study of the Variation in the Dimensions of Genuine Signatures," Journal of Forensic Science Society, Vol. 25, pp. 207-215, 1985.

[12] https://www.android.com/, Last Accessed on 12/06/2017.

[13] G. W. Bassel, E. Glaab, J. Marquez, M. J. Holdsworth, and J. Bacardit, "Functional Network Construction in Arabidopsis Using Rule-Based Machine Learning on Large-Scale Data Sets," The Plant Cell, 23 (9): 3101–3116.

[14] Hovemeyer, D. and Pugh, W., "Finding Bugs is Easy", In Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages, and Applications, 2004.