A Framework for BYOD Continuous Authentication: Case Study with Soft-Keyboard Metrics for Healthcare Environment

Luis de-Marcos[1][0000-0003-0718-8774], Carlos Cilleruelo[1], Javier Junquera[1] and José-Javier Martínez-Herráiz[1]

[1] Departamento de Ciencias de la Computación. Universidad de Alcalá. Edificio Politécnico. Campus Universitario, Ctra. Madrid-Barcelona km, 33, 600 Alcalá de Henares. Madrid. Spain
luis.demarcos@uah.es, carlos.cilleruelo@uah.es, javier.junquera@uah.es, josej.martinez@uah.es

Abstract. Mobile authentication is a hot topic because organizations can adopt BYOD (bring your own device) policies that allow to use personal devices, rather than require the use of officially provided devices. However, this brings additional access control issues like intentional or unintentional unauthorized uses of devices (e.g., stealing a mobile phone) that may eventually result in access to sensitive information. Continuous authentication (CA) aims to mitigate and provide a solution to access control by monitoring user activity. CA can then be particularly useful in mobile BYOD environments. However, each CA solution has to be implemented and integrated ad-hoc and tailored for each particular information system that wants to use it. This paper presents a modular, extensible framework for CA that enables to integrate new agents and models to implement access control with mobile devices. The framework includes three main types of components: Endpoint Detection and Response (EDR) Agents that run on the mobile device to gather user metrics and evaluate user's trust, APIs that collect information and return trustworthiness levels of users, and AI models that predict the trust of users. The framework also integrates authorized third parties that can ask for trust levels of individual users and are responsible for implementing the resulting security measures like raising alerts. The architecture is demonstrated in a healthcare environment which is part of the ProTego project. The proof-of-concept implements a mobile EDR agent and AI model based on the soft-keyboard input data collected on the mobile phone.

Keywords: Continuous authentication, access control, bring your own device (BYOD), mobile security

1 Introduction

Bring your own device (BYOD) advocates for employees bringing their own devices to work and using them to access corporate information systems, instead of using devices provided by their organization. Although BYOD offers important advantages

to both employers and employees, there also significant concerns about security and privacy [1]. Besides careful risk assessment and mitigation, contingency measures include prevention and reaction by employers and security providers [2]. On the side of the employee, it is also important to increase awareness for prevention. Prevention measures consider additional access control and authorization. Since a mobile device is something easy to take and use by unauthorized third parties, continuous authentication (CA) provides a means to monitor user activity and determine whether she is the legitimate user of the device. CA systems are those that do not require an active participation of the user to determine her identity.

CA is particularly relevant in healthcare because health-can providers can deploy BYOD policies for their employees. Also, they can enable patients to access their medical information and records or to provide data to feed these medical records through mobile apps or IoT devices, like fit bands. The ProTego project is an EU funded project that aims to provide a toolkit for health care organizations to assess better and reduce cybersecurity risks related to remote devices access to healthcare data. CA for BYOD is one of these tools.

Current literature on CA provides a variety of methods that can be applied to BYOD mobile healthcare environments. However, currently, there are no architectures or frameworks that assist when CA has to be integrated into a wider context. This is particularly important in healthcare where CA has to be integrated into heterogeneous ecosystems that include a variety of hardware and software by multiple providers, as well as data coming from multiple sources including devices and sensors [3].

This paper contributes to knowledge by providing and showcasing a CA framework for the ProTego healthcare scenario. Further, a flexible framework should also consider multiple CA methods. Since it needs to combine EDR mobile agents, APIs, authorization servers, and third parties, a scalable framework is of interest for researchers and practitioners in a variety of contexts in which they need to design and implement modular architectures that support CA. The rest of the paper is structured as follows. Section 2 summarizes the state of the art of CA. Section 3 presents the modular framework for CA and describes its components and operation. Section 4 describes a CA architecture for the ProTego project that showcases the framework in a specific healthcare BYOD setting. Finally, section 5 summarizes conclusions and future work.

2 State of the art

Continuous authentication is a procedure of access control that ensures that the identity of the user does not change during her operation of the system. The accuracy of the CA system is usually the indicator of its capabilities. There several approaches that can be based on biometrics or behavioral patterns of the user (e.g., typing). For biometric CA systems, data used ranges from the face, to finger or even electrocardiogram data [4]. For behavioral patterns, CA systems usually define metrics based on the interaction of the user with the system. Data can then be processed using different

methods and algorithms that build a prediction function that can be used to determine the probability of the user being legitimate based on future interaction. Approaches can usually be adapted between different systems even if the interaction method varies substantially. For instance, mouse pointer dynamics are similar to touchpad interaction in a mobile phone even though each of them has its particularities [5].

Keystroke and typing patterns was one of the first methods studied [6] given the ubiquity of keyboards or soft-keyboard on each interaction method that requires text input. The different metrics and mechanics of typing sessions studied include typing speed, time between keystrokes or time pressing each key [7]. For mobile devices, touchscreen dynamics drive the most significant amount of research on behavioral biometrics for CA [8] since a touchscreen is also present on most mobile devices including all mobile phones. Interactions with touchscreens also offer a lot of possibilities for exploration since users tend to interact more naturally than with a keyboard in which interaction is limited to press keys. For instance, how the user scrolls through the text provides a significant amount of data [9] that can be used to compute behavioral biometrics. Also, existing knowledge about keystroke dynamics can be combined and adapted to the touchscreen to determine how the user types in a soft-keyboard [10]. Finally, mobile devices also include a variety of sensors (e.g. accelerometer, gyroscope) that provide additional input about the user interaction. Hand Movement, Orientation, and Grasp (HMOG) [11] define a set of behavioral metrics for CA that combines traditional keystroke mechanics with sensor input. HMOG metrics can even provide additional information about the situation in which the user is interacting with a smartphone, e.g., walking or moving, sitting, or lying down. Data to feed these CA systems may come from a variety of sensors like light [12] or position sensors [13], other input and interaction methods with the smartphone like the microphone [14], or other information that can be gathered from the smartphone, like applications or processes running or geolocation.

3 Modular framework for continuous authentication

The framework (Fig. 1) includes three types of components: Clients (AI Clients & Trust Clients), Authorization Server (OAuth), and an API (ProTego JBCA). The framework can also integrate third-party systems getting trust levels, and third party systems with CA capabilities feeding the API. AI Clients have CA capabilities to get user metrics and evaluate the trustworthiness of users. User metrics are specific information or input from the user in the device that can be used to create AI models that predict the trustworthiness of a user. The authorization server secures access to the API. We suggest using OAuth2 standard with PKCE.

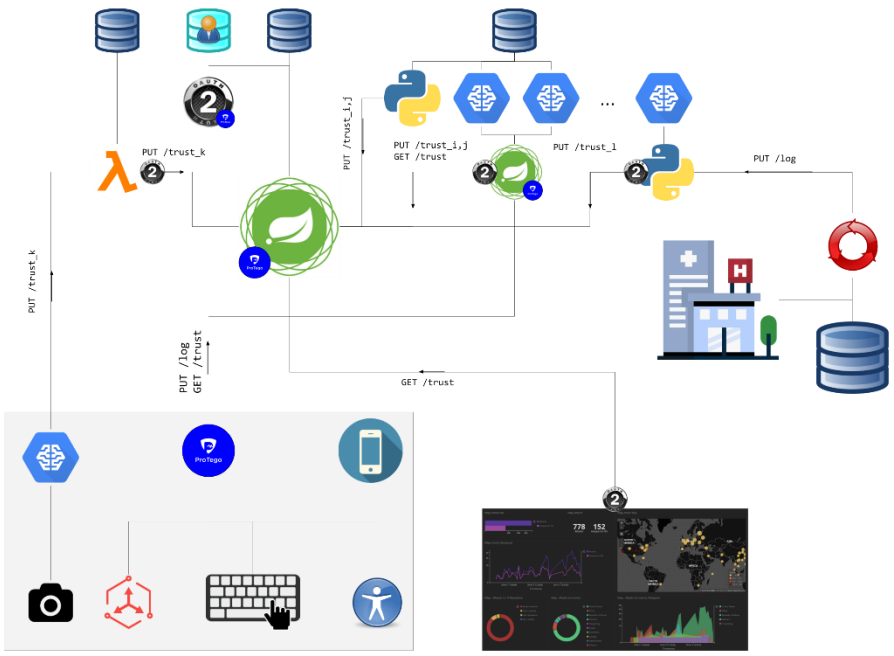

Fig. 1. ProTego Continuous authentication framework

To create an AI model, it is necessary to provide the training data and evaluation data. Additionally, it is necessary to define the expiration time of the model. The quality of each model will be based on the values of machine learning metrics. Precision should be larger than 0.5 and recall should be close to 1. To produce AI models, it is necessary to collect and train machine learning algorithms for each particular input metric (e.g., soft keyboard). Trained models can then be incorporated into the framework to provide trust levels through the evaluation of users continuously. Trust values contain a timestamp, a time of validity, the trust value (a number between 0 and 1) and an ID of the AI model. Several AI models can be used to calculate the final trust value of a user using different methods like majority voting or a weighted sum. AI clients continuously retrieve data that send to the API to produce the AI models. When models are produced and running, AI clients also retrieve new data and periodically evaluate them using the AI model. AI clients may have the capability to generate their own AI models if they have the computational capacity. However, in the first stage of the implementation of the framework, AI models are trained externally in specific servers with data gathered from EDR clients. Clients must be registered and authorized.

The API (ProTego JBCA) can be created in two different ways. In the first stage, the API and the authorization server share a common database of users and clients. When users are created, the relationship between their identity and their data is stored in the same database. The second approach is ad-hoc user creation so that each time

that a token correctly generated (i.e., made by a trust authorization server) is received, it is considered to be valid. If the user does not exist, we create it on-the-fly, associating its ID with the client (to avoid collisions). The roles considered in the API are Admin, User and Client. Admins have full access and can create clients and roles. Users can log in and manage the permissions of their clients over their information. Clients can register new AI models and manage their identities.

Figures 2 and 3 show interaction diagrams of two use cases: Authentication with the API (fig. 2), and Adding Trust Values (fig. 3). An example of global operation in which trustworthiness is decided based on multiple metrics will be as follows: (1) the AI client can access the API with an OAuth token, (2) AI clients get logs of metrics, generate AI models, and evaluate the trustworthiness of users, (3) AI clients send trust values back to the API, (4) the API combines the results from multiple sources (AI clients, third parties, …), (5) Trust clients ask the API the trust level of the user.

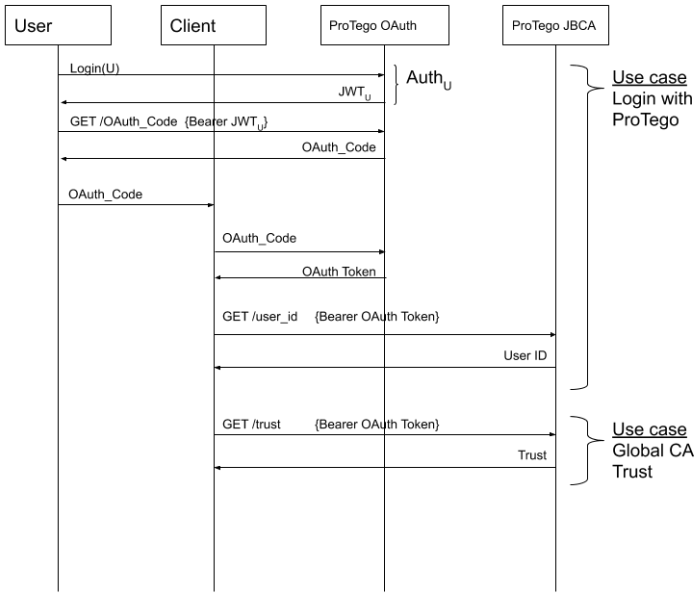

Fig. 2. Use Case: Authentication with the API (ProTego JBCA)

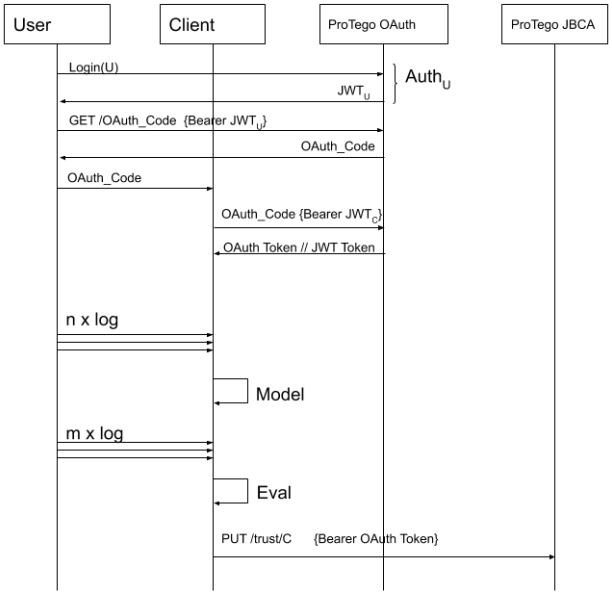

Fig. 3. Use case: AI client produces and sends trust values to the API

This framework is scalable, enabling flexible integration of new components. Specific architectural models that implement it could be designed and implemented. For instance, each mobile device could run an agent (IA agent) which reports collected metrics to the API. The API would be responsible for storing the metrics collected to be used for CA. Examples of metrics include input events in the device (e.g., keyboard, touchscreen), its sensors (e.g., accelerometer, gyroscope) or any other information that can be extracted about the individual use of the device (e.g., use of applications, processes run). The AI agents and API also evaluate the trustworthiness of the user based on AI (machine learning) models. The architecture considers the integration of authorized third party systems that can ask the API about the trustworthiness of users and decide how to act based on their own rules. For instance, third parties can send an alert when the user's trust drops below a given threshold or log the user out requiring a new authentication.

This framework offers several advantages. Its modular plugin model facilitates maintainability, scalability, and extensibility. For instance, a new AI model can be developed and integrated into the architecture without needing to add or change any existing component. Similarly, new AI agents for different user metrics or any other information that can be monitored, implemented and incorporated into the model. Please note that this includes new kinds of inputs by other devices that may be used or found useful in the future. State of the art shows a wide variety of biometric and soft-biometric methods for continuous authentication that can be integrated, or novel

methods that are not known or fully explored today can also be included. Another advantage is the integration of third-party users directly in the framework so that any authorized external source can ask about the trustworthiness of users and decide how to act, for instance, raising to alert (e.g., SIEM). So the architecture does not impose any predefined level or threshold about the trustworthiness of the user, and external can decide or fine tune the level based, for instance, in the sensitivity of data.

4 The ProTego case study for the CA architecture

The ProTego project [15] aims to provide a toolkit for health care organizations to assess better and reduce cybersecurity risks related to remote devices access to healthcare data. It includes risk assessment and risk mitigation tools, as well as methodologies and protocols for prevention and reaction. The toolkit will provide, among others, tools for risk identification and assessment both before and during the operation of health care applications, and tools to protect sensitive data and devices used to handle it. CA is implemented in the ProTego toolkit as part of the control access tools in BYOD environment. This section describes a proof-of-concept and implementation of the elements of the CA framework described in the previous section for the ProTego mobile access control scenario.

Authorized care-takers should be able to get access to medical data using their own device in a BYOD environment. BYOD requires access control, and continuous authentication offers and improves access control capabilities in mobile scenarios, for instance, in the case of stealing a device or similar unauthorized accesses. This case study (fig. 4) focuses on providing a proof-of-concept of the framework by showing an implementation of an agent that stores information about the keyboard, accelerometer, and the processes running on the mobile phone. The EDR responses include blocking the mobile device or logging-out the user in case of unauthorized access from the mobile device using an AI model that predicts the probability of being a legitimate user based on previous interaction and use of the mobile device.

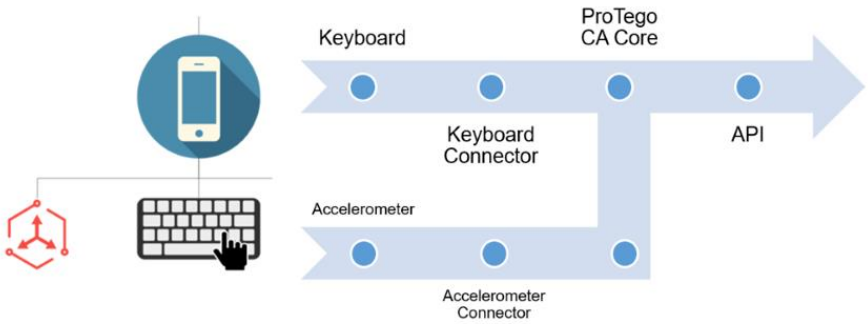

Fig. 4. Proof-of-concept of the CA framework for the ProTego case study.

4.1 Soft-keyboard biometrics

So, for this particular implementation of the framework, the AI client is an EDR mobile agent with CA capabilities to generate the trust level of a user. In this case, the agent does not need AI capabilities since AI models are trained in a backend. The EDR mobile agent component retrieves behavioral metrics and sends them to the API. Some of these metrics could be the keystroke dynamics, sensors data or gestures in the screen. When sensors generate a new log, it is sent through an internal bus to the core component (ProTego CA Core). Then, the core component processes its content to determine whether it is a continuous authentication log, and uploads to the backend. The application has been designed as a plugin model. An interface is provided that facilitates that each new feature can connect for sending logs. With this architecture, to get a new metric, it is not necessary to modify all the app, but just get the new metric and send it to the core using the interface provided. Logs are transferred as a Data Transfer Object (DTO) (fig. 5).

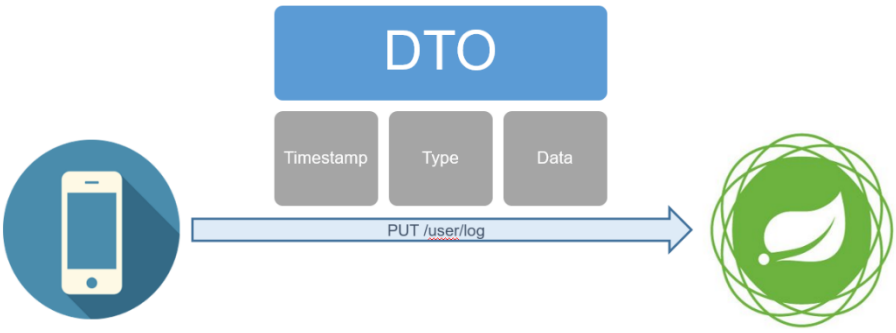

Fig. 5. Data Transfer Object (DTO) of an event log.

Mobile phone soft-keyboard events include the pressing and release of keys. These events can be used to generate a variety of metrics that identify individual users, which are used to train AI models that predict the trust level of a user. Literature of keyboard-based CA for desktop applications usually focus on the following metrics (fig. 6):

- Pressing time of each key (pressingTime)
- Time between a key release and the following key pressed (timeReleaseNextPress)
- Time between a keypress and the next keypress (timeBetweenPress)

Additionally, the codes of the key pressed can also be captured. Machine learning models eventually decide about the importance that each metric has in defining a typing pattern of the user that can be used to determine her trust level. Although the keys pressed may not seem determinant to determine identity, existing research suggests that individuals express differently and this is reflected in their wording, punctuation, etc. Since the text is composed of characters, these may be significant to deter-

mine identity. Further, although characters pressed may not be representative of a typing pattern by themselves, the combination with other metrics can provide more accurate predictive models (for instance pressing time of different characters can be different for each user). All these metrics that result from soft-keyboard interaction are gathered by the EDR mobile agent and send to the API. An example of a key event stored in the log is presented in Table 1.

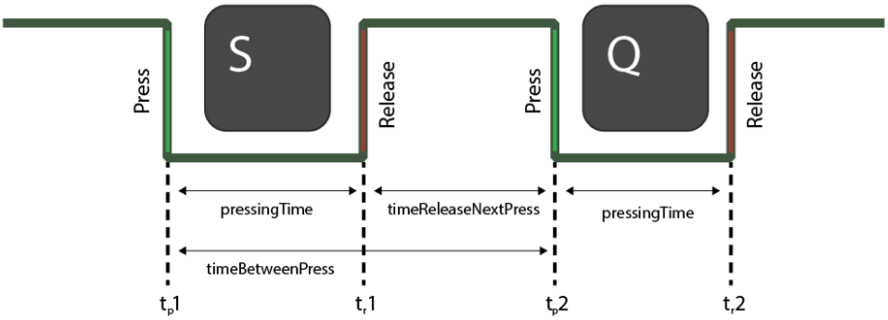

Fig. 6. Soft-keyboard events

keyCode	pressingTime	timeBetweenPress	timeReleaseNextPress	nextKey
107	53.2	143.2	90.0	101

Table 1. Example of key event log

4.2 AI models

Data of several users was collected by EDR mobile agents and used to train the AI models. Different supervised and unsupervised machine learning algorithms were selected and test for this scenario. Unsupervised algorithms included EllipticEnvelope, IsolationForest, and OneClassSVM. Supervised algorithms included RFC, SGD, and SVC. The datasets to train, test, and validate models tried to recreate real-world scenarios (Fig. 7). Particularly, the test dataset included 30% of new non-authorized user data different from the data included in the train and validation datasets. The 70:30 is a common ratio of separation used by many researchers. Implementation used a k-fold cross-validation. Initially k was established to 10.

It is an unbalanced problem since positive cases (authorized use) significantly outweigh negative cases. A negative occurs when there is an unauthorized use of a device. In most of the cases, the user is who she says she is. Because of this, we focused on unsupervised algorithms intended for use in outlier or anomaly detection and change detection (One-Class SVM, Isolation Forest, Elliptic Envelope), and on supervised algorithms that can deal with unbalanced data. The inclusion of a different test

dataset with new unused data was also intended to test the accuracy of the machine learning algorithms and its capacity to deal with the unbalance data of this particular classification problem. As future work we plan to generate and use synthetic data to further mitigate this issue.

The AI models trained can then be used by the API to answer requests of trust levels of users. In the initial prototype, the trust level is shown in a progress bar on the mobile phone (fig. 8).

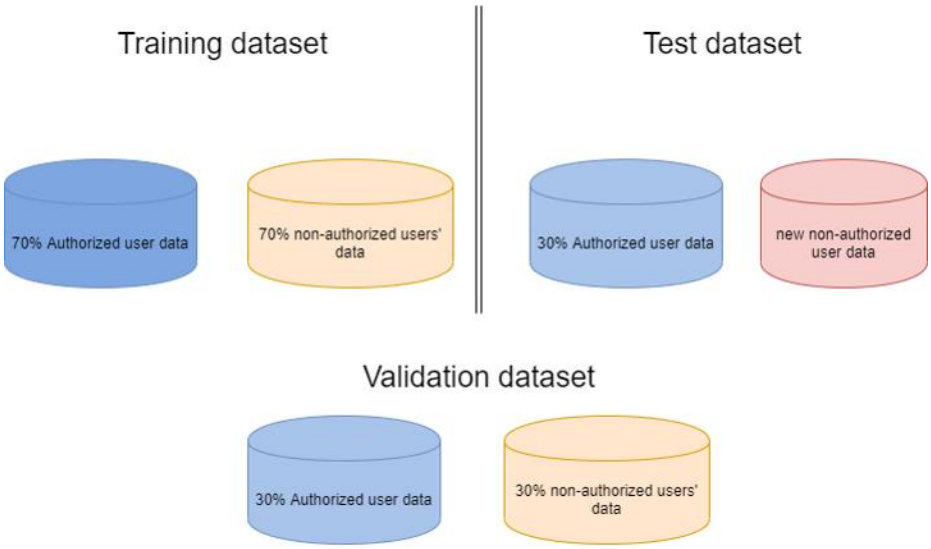

Fig. 7. Datasets for the AI model (training, test & validation)

The backend of this architecture centralizes the logs and associates them with the users' identity. All the system's information is stored in a relational database, and the logs are saved in a NoSQL database. Periodically, the backend generates the AI models using the logs stored to feed the machine learning process. The backend also evaluates new logs retrieved against these models. When an EDR agent asks for the trust of a user, the backend checks all the logs stored since the model's creation and returns a value between 0 and 1.

4.3 Next steps to complete the architecture

The components will be rearranged to improve and complete the architecture. The API will be divided in three different parts: an authorization server to manage the users, a central backend to manage the trust generated by the different AI models, and an EDR backend that would implement the AI models. It would illustrate how a third party should create his own backend acting as an AI client.

The Android mobile agent will also undertake several changes and improvements. Only the central part will be part of the integration. The EDR agent will be divided in

two components: the EDR client itself and a library for third party integration which will provide an interface that facilitates integrating new apps into the CA archirecture.

We also plan to add OpenID capabilities to the OAuth2 implementation upgrading either through the use of MitreID or through the integration in the ProTego authorization system. Another significant point is the enhancement of privacy. We are working on detecting the most appropriate data protection techniques for each scenario, as we could store a large amount of behavioral metrics. Even though in the new model the most critical data could be stored in the users' device, it can be used to generate a fingerprint of the user. We are going to study which security measures could protect the users' privacy allowing to compute with the data. These security measures over the data go from the use of secure multi-party computation to homomorphic encryption, and the evaluation of AI models under different perspectives (like differential privacy).

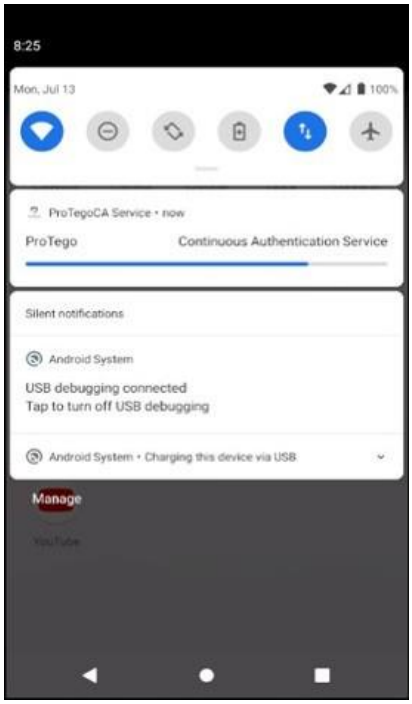

Fig. 8. Example of the trust level of the user of the CA EDR agent running in the mobile phone

5 Conclusions and future work

BYOD policies require additional access control to guarantee the identity of the users of mobile devices. CA offers this additional access control, but CA functionalities have to be included in existing information systems. This paper presented a modular framework that integrates clients, an API, authorization servers and third parties to

provide a scalable model for BYOD CA. The framework was showcased in a healthcare scenario of the ProTego project which implements CA based on soft-keyboard metrics of the mobile phone. The main components of the architecture are the EDR agents and the AI model.

As future research, the ProTego project aims to complete the architecture and toolkit to provide state-of-the-art high security in healthcare environments. Future research lines include network slicing [16] for securing wifi access inside the facilities of health care providers for users (patients and healthcare staff). Future work also includes modular encryption systems that enable transmitting and computing securely with data which can be also complemented with full memory encryption systems that enable computation with encrypted data, like homomorphic encryption [17]. Future research that is specific for continuous authentication and the framework proposed in this paper include using ensemble learning based on multiple AI models and using machine learning models based on image classification. Key events as well as touchscreen events (gestures) can be used to create heatmaps that can feed new AI models. EDR capabilities could also be augmented through malware software detection in mobile devices. Finally, the architecture will consider the presence of a local model on the user side. For instance, a model can locally generate trust levels based on a picture taken o some other local events or information. The architecture will include a local model agent to check the integration with this kind of systems.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under grant agreement No. 826284.

References

1. Miller, K.W., J. Voas, and G.F. Hurlburt, *BYOD: Security and Privacy Considerations.* IT Professional, 2012. **14**(5): p. 53-55.
2. Sequeiros, J.B.F., F.T. Chimuco, M.G. Samaila, M.M. Freire, and P.R.M. Inácio, *Attack and System Modeling Applied to IoT, Cloud, and Mobile Ecosystems.* ACM Computing Surveys, 2020. **53**(2): p. 1-32.
3. Shuwandy, M.L., B.B. Zaidan, A.A. Zaidan, and A.S. Albahri, *Sensor-Based mHealth Authentication for Real-Time Remote Healthcare Monitoring System: A Multilayer Systematic Review.* J Med Syst, 2019. **43**(2): p. 33.
4. Zhang, Y., R. Gravina, H. Lu, M. Villari, and G. Fortino, *PEA: Parallel electrocardiogram-based authentication for smart healthcare systems.* Journal of Network and Computer Applications, 2018. **117**: p. 10-16.
5. Mondal, S. and P. Bours. *Continuous authentication using mouse dynamics*. in *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*. 2013.
6. Shepherd, S.J. *Continuous authentication by analysis of keyboard typing characteristics*. in *European Convention on Security and Detection, 1995.* 1995.

7. Pisani, P.H. and A.C. Lorena, *A systematic review on keystroke dynamics.* Journal of the Brazilian Computer Society, 2013. **19**(4): p. 573-587.
8. Frank, M., R. Biedert, E. Ma, I. Martinovic, and D. Song, *Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication.* IEEE Transactions on Information Forensics and Security, 2013. **8**(1): p. 136-148.
9. Siirtola, P., J. Komulainen, and V. Kellokumpu, *Effect of context in swipe gesture-based continuous authentication on smartphones*, in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. 2018: Bruges (Belgium). p. 639-644.
10. Gascon, H., S. Uellenbeck, C. Wolf, and K. Rieck, *Continuous Authentication on Mobile Devices by Analysis of Typing Motion Behavior*, in *Proc. of GI Conference "Sicherheit" (Sicherheit, Schutz und Verlässlichkeit)*. 2014: Vienna.
11. Sitová, Z., J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K.S. Balagani, *HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users.* IEEE Transactions on Information Forensics and Security, 2016. **11**(5): p. 877-892.
12. Basar, O.E., G. Alptekin, H.C. Volaka, M. Isbilen, and O.D. Incel, *Resource Usage Analysis of a Mobile Banking Application using Sensor-and-Touchscreen-Based Continuous Authentication.* Procedia Computer Science, 2019. **155**: p. 185-192.
13. Katevas, K., H. Haddadi, and L. Tokarchuk, *SensingKit: a multi-platform mobile sensing framework for large-scale experiments*, in *Proceedings of the 20th annual international conference on Mobile computing and networking, MobiCom '14*. 2014, Association for Computing Machinery: Maui Hawaii US. p. 375–378.
14. Bonastre, J.-F., F. Bimbot, L.-J. Boe, and I. Magrin-Chagnolleau, *Person authentication by voice: A need for caution*, in *8th European Conference on Speech Communication and Technology, EUROSPEECH 2003 - INTERSPEECH 2003*. 2003: Geneva, Switzerland.
15. *ProTego: Data-protection toolkit reducing risks in hospitals and care centers. ProTego project. https://protego-project.eu/.* 10th June 2020].
16. Isolani, P.H., N. Cardona, C. Donato, G.A. Pérez, J.M. Marquez-Barja, L.Z. Granville, and S. Latré, *Airtime-Based Resource Allocation Modeling for Network Slicing in IEEE 802.11 RANs.* IEEE Communications Letters, 2020. **24**(5): p. 1077-1080.
17. Naehrig, M., K. Lauter, and V. Vaikuntanathan, *Can homomorphic encryption be practical?*, in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*. 2011, Association for Computing Machinery: Chicago, Illinois, USA. p. 113–124.