Securing Smart Homes using a Behavior Analysis based Authentication Approach

Noureddine Amraoui
Mediatron Lab
Higher School of Communications of Tunis
Ariana, Tunisia
houcine1amraoui@gmail.com

Amine Besrour
Mediatron Lab
Higher School of Communications of Tunis
Ariana, Tunisia
amine.besrour@supcom.tn

Riadh Ksantini
Digital Security Lab
Higher School of Communications of Tunis
Ariana, Tunisia
riadh.ksantini@supcom.tn

Belhassen Zouari
Mediatron Lab
Higher School of Communications of Tunis
Ariana, Tunisia
belhassen.zouari@supcom.tn

***Abstract*—This paper presents TRICA, a security framework for smart homes. When using controlling apps (e.g., smartphone app), TRICA makes sure that only legitimate users are allowed to control their Internet of Things (IoT) devices. Leveraging User Behavior Analytics (UBA) and Anomaly Detection (AD) techniques, TRICA collects and processes the historical cyber and physical activities of the user in addition to the historical states of the smart home system to build a One Class Support Vector Machines (OCSVM) model. This model is then used as a baseline from which anomalous commands (i.e., outliers) should be detected and rejected, while normal commands (i.e., targets) should be considered as legitimate and allowed to be executed. Experiments conducted on adapted real-world data properly show the feasibility of such user behavior-based authentication approach. TRICA exhibits low false accept and false reject rates ensuring both security and user convenience, respectively.**

***Index Terms*—Architecture, Internet of Things (IoT), Smart Homes, Security, User Behavior Analysis (UBA)**

I. Introduction

The Internet of Things (IoT) is becoming increasingly widespread in home environments. Consumers are transforming their homes into smart spaces with Internet-connected sensors, lights, appliances, and so on. Smart home owners are now able to manipulate their intelligent devices either from the inside, using control panels (e.g., Vivint), or from any outside location, using smartphone and web apps (e.g., SmartThings). However, compromising controlling device or security credentials could raise a serious security and privacy concerns for smart homes owners. In fact, as controlling apps only provide login-time authentication, no verification will be required once the app has been closed and then opened another time. Consequently, once compromising the controlling app, an adversary became fully privileged to control any device with no way to verify his/her identity.

Smart home users usually follow frequent patterns when manipulating their intelligent devices. Specifically, a user frequently controls a specific number of devices control in a given period of day, while following a particular order of devices. Authenticating smart home users based on these behavioral features has many advantages compared to conventional authentication schemes. In fact, user behavioral patterns could be non-obtrusively collected and monitored. Moreover, authentication could be continuously performed throughout the entire user's control session. Assessing users actions based on their previously collected activities is better known as User Behavior Analytics (UBA) [5]. Anomaly Detection is considered as the main enabler of UBA technology. AD consists of first building a baseline model over target data. Then, adherence to or deviations from this baseline could be further analyzed accordingly (i.e., accepted or rejected).

Following the user behavior-based approach and leveraging AD paradigm, this paper introduces TRICA, a security framework for smart homes. TRICA was first indtroduced in [1] and is distributed on all levels of a smart home architecture and operates in three main processes. First, the historical cyber and physical activities of the user in addition to the historical states of smart home system is collected. Then, an offline process (called enrollment) is performed to build a set of baseline models over the collected data. Specifically, a One Class Support Vector Machines (OCSVM) model is trained on a set of scores (called behavioral scores). Finally, the trained baseline OCSVM model is then saved to be used in an online

978-1-7281-5320-9/20/$31.00 ©2020 IEEE

process (called continuous authentication) to continuously verify the legitimacy of the user when controlling his/her IoT devices. TRICA stands for Trust-based, Risk-aware, Implicit, and Continuous Authentication, thus its ensures four security requirements.

The rest of this paper is organized as follows. Section 2 presents the related research literature of this work. Section 3 presents a detailed description of the core modules underlying the operation of TRICA. Section 4 presents the experimental study to show the feasibility of such user-behavior approach. Finally, Section 5 concludes this work and underlines some future directions.

II. Related Work

Researchers have been extensively working on techniques leveraging UBA and AD to secure different applications and computing systems such as Web-based Applications, Databases, and Online Social Networks (OSN).

A. Web Applications

Web logs is rich data to extract clients behavioral patterns when requesting web resources. Anomalous requests could be consequently detected based on normal requesting patterns seen in the logs. In this context, many methods has been proposed to detect security threats such as Distributed Denial of Service (DDoS) attacks. Assuming that web users access and spend more time on pages of their interests, Liao et al. extract two features to describe this behavior viz., the number of user requests in every sub-time window and the duration between them [7]. These features are then used to build a classification algorithm based on Sparse Vector Decomposition and Rhythm Matching to detect anomalous users requests. Najafabadi et al. proposed to use the Principle Component Analysis (PCA) algorithm to detect DDoS attacks [10]. The idea was to identify the N-top principal components that better describe the normal user behavior. Then, the data projection on the remaining components captures anomalies and noise in the data.

B. Databases

Databased clients usually follow some patterns when requesting DB resources (i.e., Tables). Learning these patterns allows to detect deviations that could be a sign of an abnormal access. Mathew et al. proposed a clustering-based AD technique to detect the abuse of privileges of DB users [8]. When a new query arrives, if it belongs to the user's cluster, it will be classified as normal, or abnormal otherwise. More recently, Mazzawi et al. [9] proposed to extract three behavioral features from DB audit logs, viz., (1) rarity: probability of appearance of the action in a new timeframe, (2) volume: number of occurrences of an action given that it appeared in the timeframe, and (3) new object: amounts of new objects being accessed. has been used to describe users activities. These features are then used to determine whether a DB user activity is malicious or not by assigning an anomaly scores to user actions.

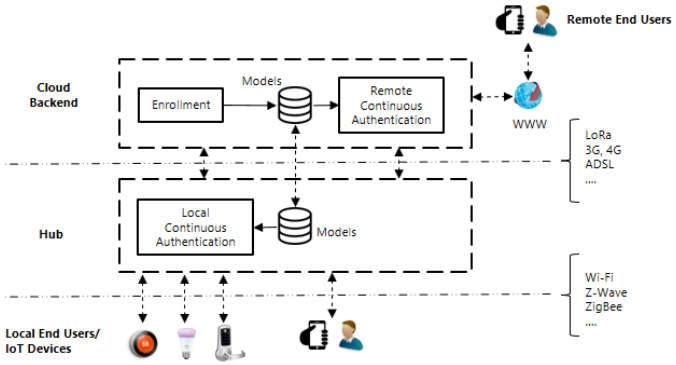

Fig. 1. High-Level Architecture of TRICA

C. Smart Homes

Nevertheless, authenticating smart home users based on their interactions with IoT devices is poorly addressed by the research community. In this work, we draws the inspiration from the afore-discussed prior works, since many behavioral characteristics such as Browsing Sequence in Web applications, Actions Rate in Databases, could also be employed in the context of smart homes to detect anomalous devices control commands.

III. Framework Design

In this section, the high-level architecture of TRICA is first introduced. Then, the framework operation is discussed in details.

A. Overview

Figure 1 shows the distribution of TRICA on the three levels of a smart home architecture. Specifically, the first step towards the building of legitimate user behavioral patterns is the collection of historical data of both user and devices on the hub (Section B.1). After that, a process called enrollment (Section B.2) is executed to build the baseline models summarizing user patterns seen in the collected data. Since the hub is the main coordinator of smart home system, the collection of data is performed on this device, while the enrollment is performed on the cloud as it is a resource-consuming task. Once the enrollment stage is finished, the local/remote continuous authentication is ready (Section B.3). In particular, if a control command is locally requested (e.g., using control panel), the analysis is performed on the hub using locally stored models. However, if a control command is remotely requested (e.g., using smartphone app), the analysis is performed on the cloud backend using cloud-stored models.

B. Framework Operation

1) Raw Logs Collection: The data used in the enrollment of user behavioral pattern consists of different types of logs that capture historical information about both users and devices. The structure of raw logs is given in Table 1.

- User Control Log: traces the history of the authenticated user when controlling his/her home devices.

TABLE I
STRUCTURE OF RAW LOGS

Log	Structure
User Control Log	Command ID, Controlled Device, Control Action, Timestamp
System States Log	Timestamp, State of Devicei, ..., State of Devicen

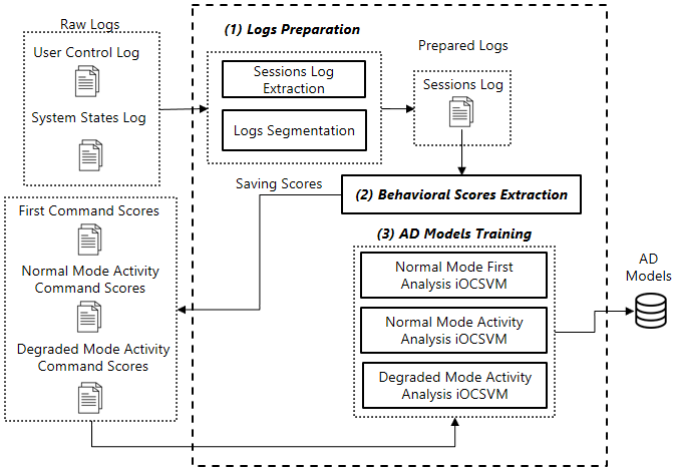

Fig. 2. Enrollment Stage

- System States Log: captures the different states of the smart home system while devices are being controlled. Each record in this log describes the status of each IoT device together with the rest of devices at a given time of the day.

2) Enrollment: Figure 2 describes the enrollment stage including four sub-processes viz., Logs Preparation, Behavioral Scores Extraction, and AD Models Training. The subsequent sections explain each one in details.

a) Logs preparation: before used in the AD models training, the collected logs need first to be prepared. This task allows the extraction of more information about the user from the raw data.

- Sessions Log Extraction: a smart home user session is the set of controlled IoT devices within a time window. Three information are used to describe a user session viz., Starting Timestamp: timestamp of the first command of the session, Control Rate: number of devices controlled during the session, and Control Sequence: order of control during the session.
- Logs Segmentation: Segmenting the logs consists of adding the corresponding time interval of the day (called period) to each row record in the logs. Adding this information gives another precision to understand the user frequent pattern. In fact, the behavior of a smart home inhabitant through the 24 hours of the day is generally segmented into a set frequent periods wherein the user has some specific behavioral routines (e.g., waking up and going to work in the morning, sleeping at night, etc.). In this work, we follow an unequal interval-based segmentation strategy in which the 24 hours are divided in 2 or more periods with unequal lengths.

b) Behavioral Scores Extraction: we call the data on which the AD models learn, the behavioral scores. These scores allow to describe devices control commands in a feature-based structure that is appropriate for a ML classifier training. In particular, this task consists of calculating a set of scores for each command seen in the control-activity log using the constructed UBM and SBM models (see Figure 2), in addition to other scores that can be calculated independently. In fact, two types of commands may be distinguished, each of which represented with a specific and common set of features as follows.

First Command Scores: five (5) features are used to describe a user behavior when starting a control session:

- Device Initialization Probability: Probability by which the user starts requesting the given device with the given action.
- Inter-Sessions Latency: Delay between this first requested command and the last command of the previous session that belongs to the same period of time.
- Control-Activity Probability: Probability by which the user requests to control the given device for a the given time period while doing the given physical activity.
- State Transition Probability: Probability by which the home system transits to the state resultant from the execution of the requested command.
- State-Activity Probability: Probability by which the home system transits to the state resultant from the execution of the requested command while the user is doing the given physical activity.
- Device Frequent State Probability: Probability by which the given device be in the given state (requested action) in the given period of time.

Activity Command Scores: Eight features are used to describe a user behavior when he/she performs a control command preceded by other commands belonging to the same session:

- Device Transition Probability: Probability by which the previously controlled device would be followed by the device requested in the given command in the same control session.
- Current Sequence Probability: Probability to see the current control sequence.
- Intra-Session Latency: Delay between the requested control command and its antecedent in the same session.
- Control Rate: Number of current controlled devices.
- Control-Activity Probability: Probability by which the user requests to control the given device for a the given time period while doing the given physical activity.
- State Transition Probability: Probability by which the home system transits to the state resultant from the execution of the requested command.
- State-Activity Probability: Probability by which the home system transits to the state resultant from the execution

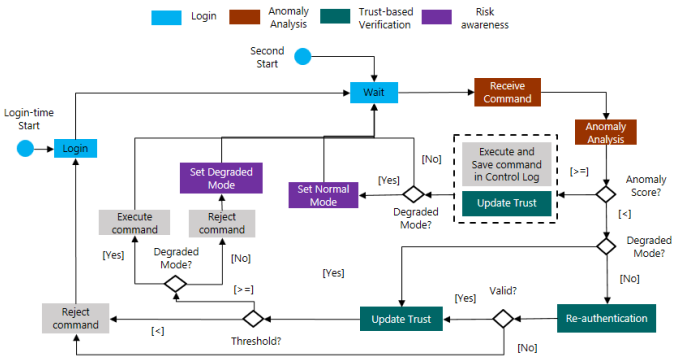

Fig. 3. Continuous Authentication Workflow

of the requested command while the user is doing the given physical activity.

- Device Frequent State Probability: Probability by which the given device be in the given state (requested action) in the given period of time.

c) Anomaly Detection Models Training: training the AD models on the set of extracted normal behavioral score is the fruit of all the previous enrollment sub-processes. Since our objective is to discriminate legitimate control commands from anomalous ones, we are dealing with a binary classification problem. However, as only target behavioral scores are available during the enrollment stage, One Class Classification (OCC) should be used in such situation. In this work, we use the One Class Support Vector Machines (OCSVM) [12] as it has shown high performances in detecting anomalies in many other application domains compared to other AD algorithms [4]. In particular, as two types of control commands are distinguished viz., first and activity command, two OCSVM models (called F-OCSVM and A-OCSVM, respectively) are trained each on the corresponding extracted behavioral scores.

3) Continuous Authentication: Figure 3 describes the workflow of the Continuous Authentication, including four (4) inter-complementary procedures (i.e., each with different color; functions belonging to the same procedure have the same color):

- Login (in blue): by providing the correct e-mail and a password, the user should be authenticated to the controlling app and authorized to control the home devices. However, if the controlling app has been closed and opened once again, no login is required. The wait function in the figure means that the continuous analysis is only triggered by user requested commands.
- Anomaly Analysis (in brown): upon receiving a requested control command, the anomaly analysis sub-process is triggered. Specifically, the corresponding behavioral scores of the requested command are first calculated according to its nature (i.e., first or activity command). Then, the corresponding AD OCSVM model is used for analyzing the calculated behavioral scores (i.e., F-OCSVM, A-OVSVM) outputting an anomaly score. If this score is above a predefined threshold, the requested command is indicated as legitimate and then executed and saved in the control log. However, a score below the threshold indicates an abnormal command that triggers another verification.
- Trust-based Verification (in green): if the requested command is analyzed as suspicious, an explicit re-authentication (e.g., password verification) is prompted to the user. In order to follow the confidence level related to user behavior, we leverage the trust-based verification first introduced in [2]. In particular, a trust value is calculated using the anomaly analysis score (aas) outputted from analyzing the anomaly of the requested command. Hence, the confidence towards the current user may increase or decrease according to this score. This technique allows to prevent a user, that can successfully pass the re-authentication while continuously showing an anomalous behavior, from keep using the controlling app. Precisely, if the user fails to re-authenticate, the requested command is directly rejected and the user is completely logged out from the main interface of the controlling app. However, if the re-authentication is valid, the user confidence is verified. If user trust is still below the allowed level, the command is executed. Yet, if user trust drops below the threshold, the requested command is rejected, and the user is logged-out.
- Risk-awareness (in purple): if the user is able to re-authenticate and shows an acceptable level of trust, the requested commands are executed, but his/her behavior is still considered as suspicious. In order to reduce the impact of suspicious executed commands during the trust-based verification, a degradation to a restricted control mode is proposed. This scheme makes the continuous authentication aware of the risk that may come with false executed commands, as the legitimacy of current user is not certain. Precisely, when switching to degraded mode, high-sensitive devices are disabled, and the current user can only request to control low-sensitive devices. The switch-back to the normal mode of control is done when the user request a command analyzed as legitimate, meaning that the current user has shown a normal behavior.

IV. Experimental Study

In this section, we validate the generality of TRICA in detecting anomalous user behavior on datasets that involve different smart home environments. Specifically, we first present the description of the evaluation dataset. Then, we introduce the evaluation methodology. Finally, we discuss the obtained experimental results.

A. Dataset Description

As it has been extensively used in many ambient intelligence applications such as home recommender systems [11], user physical activities history inside a smart home environment can be easily found in public repositories. However, the history data of control and states of smart home IoT devices when they are controlled by their users is not publically available.

To remedy to the lack of such data, we propose to adapt the historical data of manual control of appliances and objects by inhabitants in real-world home environments, and assume that they refer to app-based control of home IoT devices.

The data we will be using is the one collected by the University of Amsterdam [6] which contains three datasets (called house A, B, and C). Each one of the houses was instrumented with wireless sensors (e.g., contact switches to measure open-close states of doors; pressure mats to measure lying in bed, etc.) to record the activities of one single inhabitant during several weeks. The activities include both manual control of different house appliances (e.g., open door, turn-on microwave) as well as daily living activities (e.g., cooking, sleeping, etc.).

B. Evaluation Methodology

As the evaluation dataset contains the data of three inhabitants each in one home environment, three baselines are built over the data of each one of the three inhabitants. In order to evaluate the ability of TRICA in discriminating anomalous user behavior from genuine one, we follow a primary-vs-adversary strategy. First, one inhabitant is designated as the primary user where his/her iOCSVM models are considered for the evaluation, while the two remaining inhabitants are considered as adversaries where their control commands behavioral scores are considered as the testing data. Then, a part behavioral scores of the primary user is also included in the behavioral scores of the two adversaries. This process is then repeated, designating each of other inhabitants as the legitimate user in turn.

C. Results and Discussion

Table 5 shows the mean value of the obtained measurements for the FAR (False Accept Rate) and FRR (False Reject Rate) metrics. We can see that among all possibilities of the primary-vs-adversary strategy, the FAR reaches at worst 5.84% while successfully reaching down to 0.01%. This low rate makes sure that the system is efficiently not accepting adversaries thus ensuring a high security level. On the other hand, the FRR reaches at worst 7.94% while successfully reaching down to 6.45%. This low rate ensures a high level of user convenience since legitimate user is rarely prompted to re-authenticate.

TABLE II
AVERAGE OBTAINED VALUES OF FAR AND FRR

	Primary Users					
	User 1		**User 2**		**User 3**	
Adversaries	**FAR**	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**
User 1	NA		0.0355	0.0718	0.0584	0.0718
User 2	0.0001	0.0794	NA		0.0227	0.0794
User 3	0.0144	0.0645	0.0226	0.0645	NA	

V. CONCLUSION

Smart home users tend to follow different behavioral patterns when manipulating their intelligent IoT devices. This assumption is the working principle of TRICA: a security framework for authenticating smart home users. Experimental results conducted on adapted real-world data have reinforced the ability of TRICA to differentiate between different smart home users. In the future, we plan to investigate how the normal OCSVM models should be updated to cope with the change of normal user behavior using the incremental version of OCSVM. In addition, as user behavior is being analyzed, anomalous users patterns may become available from rejected commands. We also plan to leverage such adversarial behavior to enhance authentication performance.

REFERENCES

[1] Amraoui, N., Besrour, A., Ksantini, R., & Zouari, B. (2019, March). Implicit and Continuous Authentication of Smart Home Users. In International Conference on Advanced Information Networking and Applications (pp. 1228- 1239). Springer, Cham.

[2] Bours, Patrick. "Continuous keystroke dynamics: A different perspective towards biometric evaluation." Information Security Technical Report 17.1-2 (2012): 36-43.

[3] Eberz, Simon, et al. "Evaluating behavioral biometrics for continuous authentication: Challenges and metrics." Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM, 2017.

[4] Garcia-Font V, Garrigues C, Rifà-Pous H (2016) A comparative study of anomaly detection techniques for smart city wireless sensor networks. Sensors J 16:868

[5] https://www.gartner.com/reviews/market/user-and-entity-behavior-analytics

[6] van Kasteren, Tim LM, Gwenn Englebienne, and Ben JA Kröse. "Human activity recognition from wireless sensor network data: Benchmark and software." Activity recognition in pervasive intelligent environments. Atlantis Press, 2011. 165-186.

[7] Liao Q, Li H, Kang S, Liu C (2015) Application layer DDoS attack detection using cluster with label based on sparse vector decomposition and rhythm matching. Security and Communication Networs J 8:3111-3120

[8] Mathew S, Petropoulos M, Ngo H Q, Upadhyaya S (2010) A data-centric approach to insider attack detection in database systems. In: 13th International Workshop on Recent Advances in Intrusion Detection, Berlin, pp 382-401

[9] Mazzawi H, Dalal G, Rozenblat D et al (2017) Anomaly detection in large databases using behavioral patterning. In: 33rd International Conference on Data Engineering, San Diego, CA, pp 1140-1149

[10] Najafabadi M M, Khoshgoftaar T M, Calvert C , Kemp C (2017) User behavior anomaly detection for application layer DDoS attacks. In: 18th International Conference on Information Reuse and Integration, San Diego, CA, pp 154-161

[11] Rasch, Katharina. "An unsupervised recommender system for smart homes." Journal of Ambient Intelligence and Smart Environments 6.1 (2014): 21-37.

[12] Scholkopf B, Platt J, Taylor J S et al (2001) Estimating the support of a high-dimensional distribution. Neural Computation J 13:1443–1471