# Strengthening Access Control in case of Compromised Accounts in Smart Home

Rath, Thavy Mony Annanda; Colin, Jean-Noël

*Published in:*
2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2017

*DOI:*
10.1109/wimob.2017.8115827

*Publication date:*
2017

# Strengthening Access Control in case of Compromised Accounts in Smart Home

Annanda Thavymony RATH
*Precise, Faculty of Computer Science*
*University of Namur, Belgium*
*Email: rath.thavymony@unamur.be*

Jean-Noël Colin
*Precise, Faculty of Computer Science*
*University of Namur, Belgium*
*Email: jean-noel.colin@unamur.be*

***Abstract*—Smart home user usually controls smart devices through smart application, which is managed by user's account. Thus, compromised account is possible and countermeasure to such attack can help protect both devices and data pertaining to them. In this paper, we propose a security countermeasure in case of compromised account in smart home system by introducing another layer of access control beyond the traditional authentication method (e.g. username and password). In our proposed approach, although user is successfully authenticated, he subjects to another control at devices or data permission level for every access attempt to them. This control takes into account the profile and behaviour of user requesting access to the system to determine whether user is legitimate or malicious and access control permission and type of access control enforcement are decided based on that factor.**

***Keywords*-Account hacking; access control; smart home; IoT; security countermeasure; compromised account;**

## I. Introduction

Among 8 categories of threats [5] in Internet of Things (IoT), the attack such as account hacking is an obvious cause for concern since IoT users generally use smart application to manage and control devices or data. If user's account is compromised and once attacker get user's credentials, he can use to bypass security filters and steal information, alter data, damage devices and deny service. Thus, in IoT environment and smart home in particular, it is important to put more control on devices and data even after user is authenticated.

Detecting the abnormal behaviour of user at authentication level is not new and it has been addressed and implemented in many well-known systems, such as Google Gmail or Facebook [1]. However, in such systems, once user is authenticated, there is no further control on what user is accessing or doing afterward. This security loophole allows malicious user to gain access to data without any limitation once he passes the authentication phase. Compromised account in mailing system or social network may be less severe compared with IoT services such as smart home. Hacking into cameras installed in home, violating privacy, and accessing content (pictures and movies) are some of the security threats introduced by the new era of connected homes. These violations of accessing the content of home automated devices can lead to many dangerous outcomes, such as burglary or any other form of troubles. Therefore, the issue of compromised account and its countermeasure should be addressed carefully since IoT promotes the openness and collaboration where information is more exposed to attacks compared with closed system [5].

In our security countermeasure approach, although user is successfully authenticated, he still is subjected to a denial access if the system detects that he is highly likely a malicious user. The method to find out if user is a malicious or legitimate is to analyse user's present behaviour against his past behaviour (user's access pattern), hence, user's access history is used as a source for information mining. The machine learning method such as association rule learning [8] is used for analysing the user's behaviour for user's frequential pattern. In this approach, devices and data are controlled by access control policies where the behaviour of user is expressed in these policies in the form of probabilistic value (degree of certainty whether user is legitimate or malicious). The access control model used in this paper is our proposed extended version of Attribute-based Access Control (ABAC) [7]. An access control architecture supporting our security countermeasure and its implementation in XACML [10] are also presented in this paper.

This paper is structured as follows. Section II talks about the motivation and security issues in smart home scenarios. Section III focuses on security countermeasures against compromised account. The user's profile and behaviour analysis are also discussed in this section. Other important point in this section is the mining of user's access history for frequential access pattern. The proposed access-log structure for smart home system is also discussed in this section. Section IV presents the access control model and policies expression for some scenarios in smart home. Section V focuses on access control system architecture and its implementation in XACML. Section VI is the related work and Section VII is our conclusion.

## II. Motivation and scenario

In the existing smart home services providers, users need to have accounts at clouds services, then, through this account, user can add smart devices and connect them to cloud services. After that, user can access and control them. Since user's account is generally used for managing and accessing devices,

[1] https://www.facebook.com/help/loginapprovals

978-1-5386-3839-2/17/$31.00 ©2017 IEEE

the issue of compromised account can not be ruled out. We provide account hacking scenario below and illustrate the importance of security countermeasure against such issue in smart home.

We take a smart home scenario where home owner (user) routinely executes commands to smart devices using smart home application. Each user has an account, which can be used to access the smart devices. User can, for example, turn-on or off devices (e.g. refrigerator, TV, ...), check the video surveillance and other sensing devices. The interaction between user and smart devices has been happening in the more or less precise time interval, which creates an access pattern. Suppose that there is a situation where account is hacked and malicious user executes commands on devices in the strange way (e.g. turn-off refrigerator or CCTV when user is not at home ). User has never executed such command in such situation (not at home). It is worth noting that turn-off refrigerator or CCTV has bad consequences from foods spoilage to security issue given CCTV used to surveil home remotely is turn-off. In such situation, how system intelligently react to such strange behaviour and prevent malicious user from executing those commands?

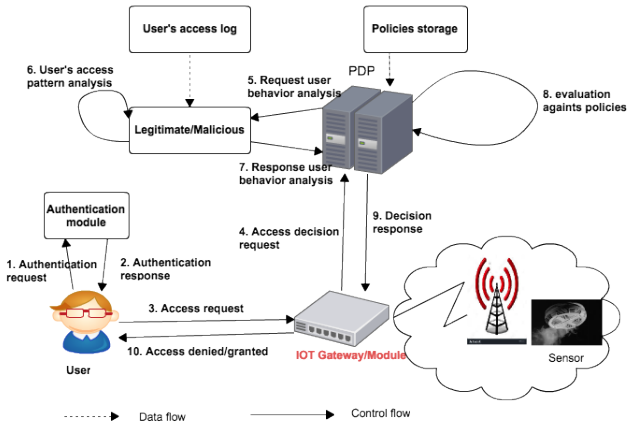


Figure 1. smart home architecture with authentication and access control supporting mechanism against compromised account

## III. Security countermeasure against compromised account

Unlike other systems [1], users, in smart home, tend to repeat their daily activities in more or less precise time interval. For example, every Monday to Friday at 6 PM, home owner arrives, opens the door and turns on air-conditioner. At 7 PM, he turns on TV and turns it off at 11 PM before going to bed. Before going to work, at 7 AM, the CCTV is turn on for surveilling the house during his absence. The CCTV is always on when home owner is not at home. All these activities (actions on devices) are more or less repeated daily and they constitute the access patterns of those devices. These access patterns can be used to prevent any abnormal access performed out of the observed patterns. For example, if an attacker successfully hacks the system, gets into user's account and commands to turn off the CCTV at the time when home owner is not at home, this action can be considered as suspicious because this pattern user has never done before. Thus, more comprehensive and restrictive access control needs to be enforced as the precaution measure. In other words, access control to devices and data must be strengthened.

### A. Smart home architecture

Figure 1 is our proposed physical architecture of smart home system where the authentication and access control modules are integrated. In this architecture, we propose to use the centralised approach [12] for access control in stead of decentralised one [12] because analysing user's access log as well as validating access control policy requires big memory and high computational power. Thus, decentralised access control approach where policy validation and decision needs to be done at device level is not suitable to be used in our case given that most smart devices have limited power and memory. In this architecture, Figure 1, there are two main parts: the user authentication and access control. The architecture consists of the following modules.

1) User is a physical person or application requesting access to devices or data pertaining to them.
2) Authentication module is responsible for authenticating user. User's credential or other information such as username or password are used for authenticating user.
3) IoT Gateway module is responsible for processing data from smart devices and communicating it to IoT platform. IoT gateway can handle different network technologies deployed at home environment, such as WiFi, Lora, Zigbee, 6LoWPAN or other wireless communication technologies. IoT supporting platform generally refers to IoT server, a module responsible for processing or storing data from smart devices and also providing services to end-user application. There are two types of IoT server: public and private servers. Public server refers to a server being publicly accessible and used by many users. For example, a cloud service or other type of similar service. For private server, it is built for a particular smart home network and it is accessible only for that network users. The private server can be installed in or out side the home network.
4) Policy Decision Point (PDP) is an access control module being responsible for filtering access requests based on their defined authorisation policies. This entity could be instantiated by or embedded into a gateway with direct communication to the devices that it manages, or another entity in a different location, either in home network or in the cloud service (see Figure 1).

5) Legitimate/Malicious is a module responsible for analysing and extracting user's access pattern based upon which the decision of whether user is legitimate or malicious is performed. Once user type is determined, the user's information is passed to PDP.
6) Home network environment consists of a network of interconnected smart devices used in home.

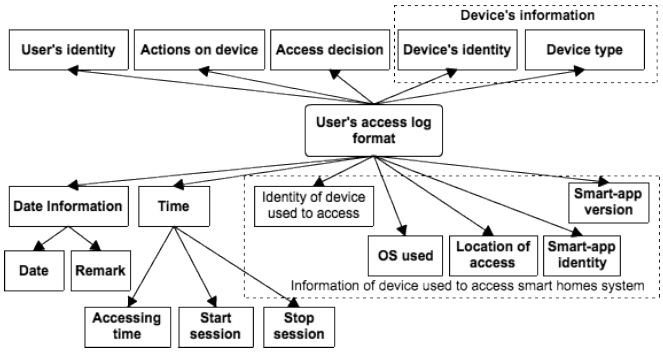


Figure 2. User's access log format for smart home system

The step-by-step information flow is presented below.

*Figure 1:* step-by-step information flow explanation

(1) Authentication request, user provides the authentication information, such as username, password or other user's credential for authentication purpose. (2) Once user is authenticated, an authentication response is sent to user. If response is positive, user can get into his account. (3) User can use available services in smart application. Every time user executes command, an access request is sent for validation. (4) Access request is sent by user to PDP for validation. (5) PDP sends a request to "Legitimate/Malicious" for user behaviour analysis. (6) "Legitimate/Malicious" analyses user's access log for access pattern. (7) After analysing user's access log, "Legitimate/Malicious" module sends user's behaviour analysis response to PDP. (8) PDP validates user's request against defined access control policies. (9) Decision response is sent by PDP to IoT gateway. (10) IoT gateway forwards decision response to user.

### *B. User behaviour analysis*

User behaviour analytics (UBA), as defined by Gartner [8], is a cyber security process about detection of insider threats and targeted attacks. UBA solutions look at patterns of human behaviour, and then apply algorithms and statistical analysis to detect meaningful anomalies from those patterns-anomalies that indicate potential threats.

Since our main objective is to determine if user is legitimate (actual) or malicious, we use UBA technique to address this issue. We apply association rule learning [8] for analysing the user's access history to extract the user's usage pattern and use this pattern to detect attack.

*1) Access log structure:* The smart home access Log is a type of log that contains all requests to resources (devices and data pertaining to them) of a smart application. For example a user requests to turn on or off smart door, CCTV or thermostat, the smart application and smart home system validates user's request and the access log will record the requests of user to devices. In our security countermeasure approach against compromised account, user's access log plays as an important role because it is the source of information for user's access pattern extraction. Thus, we define, in this section, a general access log structure for smart home system taking into account different parameters, such as spacial and temporal constraints, type of sensing devices and other contextual information. This access log structure (see Figure 2) will be used for user's behaviour mining in the following section. Our proposed access log format in Figure 2 consists of the following parameters.

- "User's identity" stores the identity of physical person or application accessing resources (devices).
- "Actions on device" stores the operations on smart devices or sensing devices.
- "Access decision" stores the information concerning the decision by PDP on user's request. The decision can be "Deny" or "Permit".
- "Device's identity" stores the unique identity of device.
- "Device's type" stores information concerning type of devices. For example, sensing device or actuator.
- "Date information" stores information concerning the date at which user accesses to smart home system. This parameter is divided into two parts: the date stores actual date (e.g. DD/MM/YYYY) and remark stores information concerning date, such as working day, weekend or holiday.
- "Time" records the time at which user accesses and uses the system. It is divided into three parts:
  - "Accessing time" stores time at which user accesses system.
  - "Start session" stores time at which the usage session starts after access permission is granted.
  - "Stop session" stores time at which the usage session stops.
- "Identity of device used to access" records the unique identity of device used to access system. For example, user uses smart phone, tablet, desktop or laptop to get access to system.
- "OS used" stores information concerning the operating system (OS) used by the device to get access to system.
- "Location of access" stores information concerning the location of user at the time of access. For example, network identity, physical or geolocation of user.
- "Smart-app identity" stores the unique identification of

smart application used to connect to smart home system.
- "Smart-app version" stores the information concerning the version of smart application used to connect to smart home system.

*2) Mining of user's access behaviour for frequent pattern:* in smart home context, there is always the relationship between commands user executes on devices and time at which they are executed. It relates to daily activities of home owner. For example, home owner turns on TV at 7 PM or smart door is open at 6 PM when he arrives home. These habit, from time to time, constitutes an access pattern. Any action that is different from user's access pattern may be questionable (e.g. account may be compromised) and more control procedure and enforcement are required.

In our proposed approach, this access pattern is important for detecting the abnormality in user's behaviour. To determine the relationship between user, devices and time at which he executes commands, we use, association rule learning, a rule-based machine learning method for discovering interesting relations between entities in smart home context (see parameters in user's access log format).

**Association rule learning (ARL)** is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules [8] discovered in databases using some measures of interestingness [8]. ARL is generally used to analyse the relationship between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. However, it is also used in other areas, which require to determine the relationship between entities in the database.

**Definition**: let X and Y be two set of entities (or item sets) where X ∩ Y=0. X ⇒ Y is an association of X and Y. Let S be a set of transactions of a given database.

**Confidence of rule** $(X \Rightarrow Y)$ is an indication of how often the rule has been found to be true.

The confidence value of a rule, $X \Rightarrow Y$ , with respect to a set of transactions S, is the proportion of the transactions that contains X which also contains Y.
Confidence is defined as: $\mathrm{Conf}(X \Rightarrow Y) = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(X)}$

Support (or supp) is an indication of how frequently the itemset appears in the dataset. The support of X with respect to S is defined as the proportion of transactions "s" in the dataset which contains the itemset X. $\mathrm{supp}(X) = \frac{|\{s \in S; X \subseteq s\}|}{|S|}$

**Apply ARL for user's access pattern extraction.** Suppose we have two sets of entities X and Y. X contains (user, action, device) and Y contains (time) (see Figure 3). We want to find the relationship between user performs action on device and time that action is executed. Since rule confidence is an indication of how often the rule has been found to be true, this means that, the high confidence indicates user's habit, hence, it can be considered as access pattern. Thus, any access request falls out of this access pattern is considered as suspicious and needs further scrutinise.

**Example 1: Rule confidence calculation**

Suppose that we have the transactions like in Figure 3[2]. User, Edward, through his account in smart home application, executes command "turn-off" to device (CCTV) at 6PM. With this request, analyse if the user is actually "Edward". Calculate the rule confidence of ((Edward, turn-off, CCTV), 6PM) given his past access history in Figure 3. $Conf((Edward, turn-off, CCTV) \Rightarrow 6PM) = \frac{\mathrm{supp}((Edward, turn-off, CCTV) \cup (6PM))}{\mathrm{supp}(Edward, turn-off, CCTV)} = \frac{3}{3}$.

With the rule confidence equals "1", the system concludes that the user is high likely "Edward".

There is one drawback for this approach that is we need to have a reasonable size of access history in order to produce a reasonable prediction. This approach can not be applied in case of new user without access history.

It is worth noting that the user's access pattern represented by the rule confidence is one of the constraints in access control policy. This access constraint along with its enforcement technique are expressed in access control policy and will be evaluated at policy evaluation phase. We present the access control model and how to express user's access pattern in that access control policy in next section.

| Transaction | User | Action | Device | Time |
|---|---|---|---|---|
| 1 | Edward | Turn-off | CCTV | 6PM |
| 2 | Edward | Turn-off | CCTV | 6PM |
| 3 | Edward | Turn-off | CCTV | 6PM |
| 4 | Charlie | Trun-off | Refrigerator | 10 AM |
| 5 | Edward | Open | Door | 5:30 PM |
| 6 | Edward | Open | Door | 5:30 PM |
| 7 | Edward | Open | Door | 5:30 PM |
| 8 | Edward | Open | Door | 9PM |

Figure 3. Example: simplified version of user access history

## IV. Access control model and policies expression

The required user's access pattern can be incorporated with other existing access control models [7] [9]. However, in this paper, we extend attribute-based access control to support our proposed approach. The idea of extending this model is that ABAC is well known access control model that is being used widely in many systems [7]. ABAC is well known for its ability to express fine-grained and complex

[2]This is a simplified version of access-log, derived from Figure 2. In this example, accessing time is used as one of entities for extracting user's access pattern. It is worth noting that in practice, it is highly likely that user could not operate, everyday, at exact point in time. For example, everyday at 6 PM , turn-on the TV. It may differ in seconds or minutes. Thus, pre-processing or mining of raw access-log is required before giving as inputs to ARL in order to achieve correct estimation.

policies and it is easy to understand and manage. The well known existing access control engine such as XACML is also implemented based on ABAC model. ABAC is good to be used in large number of IoT access scenarios in the domains, such as smart home, traffic control, agriculture and transportation [2].

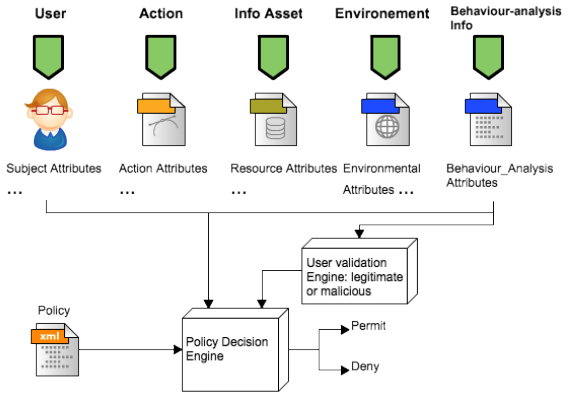


Figure 4. Extended ABAC

### A. *Extended attribute-based access control*

ABAC [7] defines an access control model whereby access rights are granted to users through the use of policies which combine attributes together. ABAC is becoming well known and considered as a "next generation" authorisation model because it provides dynamic, fine-grained, context-aware and intelligent access control. ABAC uses attributes as building blocks in a structured language that defines access control rules and describes access requests. Attributes are sets of labels or properties that can be used to describe all the entities that must be considered for authorisation purposes. As shown in Figure 4, the traditional ABAC model consists of 4 main entities (e.g. subject, action, resource and environment ) where each entity may hold multiple attributes.

- Subject attributes describe the user attempting the access e.g. age, clearance, department, role, job title...
- Action attributes describe the action being attempted e.g. read, delete, view, approve...
- Resource (or object) attributes describe the object being accessed e.g. the object type (medical record, bank account...), the department, the classification or sensitivity, the location...
- Contextual (environment) attributes deal with time, location or dynamic aspects of the access control scenario. For more details about ABAC, see in [7].

To formally incorporate "user's behaviour" into ABAC model, we propose the extension like shown in Figure 4. The user's behaviour analysis entity is responsible for providing the estimation value of whether user is considered as malicious or legitimate. The new extended ABAC policy should contain not only user, action, resource and environmental attributes, but also the user's behaviour attribute(s).

The user's behaviour information is considered as separate entity from environment attributes in ABAC model because in traditional ABAC, environment attribute is any information regarding the context of the access that might be used in making the access decision, such as, time, network or spacial context whereas user's behaviour information is the information from different sources used for estimating whether user is a legitimate or malicious. In most cases, information used for analysing user's behaviour is a complex data sets that generally come from databases (e.g. access history) or external information system.

**Extended ABAC policy expression**

- Let U be a set of users (u);
- Let A be a set of actions (a);
- Let C be a set of resources (c);
- Let E be a set of environment attributes (e);
- Let B be a set of user's behaviour analysis variables (b). This variable expresses the level of certainty if user, who is requesting an access, is legitimate or malicious. "b" is expressed in probabilistic value;
- Let O be a set of obligations (o) that user or system needs to perform if abnormal behaviour is detected.

The permission assignment in ABAC is expressed as (u, p) where p is a permissions on resource. p=(a, c, e). The behaviour-based ABAC policy expression is as follows.

p= ((a, c, e), (b , o))

---

*Definition 1:* ABAC environmental variable expression

---

Let E be a set of environmental attributes (e), where $e \in E$ . "$e$" has the finite domain of possible values, denoted as N where $n \in N$. "e" is equipped with the relational operators (Oprs) " $=, \neq, \geq, and \leq$ ". The condition of "e" has the form ($e$ opr $n$). let $e_1$ and $e_2$ be two environmental variables. Then, ($e_1 \wedge e_2$) or ($e_1 \vee e_2$) are multiple environmental variables conditioned in policy. For example, time $\geq$ 20:10:00.

---

*Definition 2:* User's behaviour expression

---

Let B be a set of user's behaviour (b), where $b \in B$. "b" has the finite domain of possible values, denoted as D where $d \in D, d = [0, 1]$. "b" is equipped with the relational operators (Oprs) " $=, \neq, \geq, and \leq$ ". The user's behaviour of $b$ has the form ($r$ opr $d$).

let $b_1$ and $b_2$ are two behaviour variables. Then, ($b_1 \wedge b_2$) or ($b_1 \vee b_2$) are multiple behaviours conditioned in policy.

---

*Definition 3:* Obligation expression

---

Let O be a set of enforcement obligation variables (o), where

$o \in O$ . "$o$" has the finite domain of possible values, denoted as B where $b \in B$. "o" is equipped with the relational operators (Oprs) " $=, \neq, \geq, and \leq$ ". The obligation of "o" has the form ($o$ opr $b$). For example, user notification obligation has the form: $notify = true$.

### B. Example: Strengthening Access Control in case of Compromised Account

In this section, we provide a complete smart home access scenario, the policy expression for controlling access to smart devices and show how countermeasure against compromised account can be achieved. We take the smart home scenario in Section II. Edward, who is the home owner, installed a smart home system in his property. All devices, including smart door, in his house can be controlled by smart application through his account. Edward's account is authenticated by username and password. Edward uses a smart door, which is controlled by the following access control policies.
(1) Policy1: rule states that user "Edward" can open smart door if the probability of certainty that user is "Edward" is greater than or equal 90%.
(2) Policy 2: the second rule state that "Edward" can open smart door if the probability of certainty that user is "Edward" is less than 90% given that Edward can answer the question he registered at the time of creating his smart home account. A notification to Edward is also required.
Suppose that b is user's behaviour analysis variable, hence, we can express the two policies in ABAC as follows.

- Policy 1: (Edward, ((open, smart-door, b $\geq$ 0.9)))
- Policy 2: (Edward, (open, smart-door, b<0.9), prove(question)=true $\wedge$ notify(Edward)=true)

In this example, policy 1 allows to open the door without any further control if the system analyses Edward's access log and finds that user, requesting to open the door, is highly likely "Edward" with the probability of certainty greater than or equal 0.9. For the second policy, if the probability is below 0.9, system requires user to prove himself by answering a question and a notification is sent to Edward for validation. The question and its answer are registered at the time Edward created account. The second condition is an enforcement operation performed by system to prevent attacker from successfully executing commands. It is important to note that the threshold "0.9", in this example, can be determined based on the observation of user's access history for a reasonable periods of time.

Suppose that an attacker successfully hacked to Edward's account and commands to open the door at 4 PM. Given Edward's access history in Figure 3 and two policies above, what is the decision of the system? The request provided to system by hacker is: (Edward, open, door, 4PM). System analyses Edward's access log and find that Conf ((Edward, open, door) $\Rightarrow$ 4PM) = 0 (see Section III.B). This means that the policy 2 is applied and attacker needs to answer the question. Since attacker does not know the answer to the question that Edward registered at the time of creating account, system rejects the request of attacker and notifies Edward about the request although attacker successfully hacked Edward's account (account is compromised).

It is worth noting that other factor that can be used for access control decision is the behaviour of user at the time of request such as number of failed access attempts.

## V. Access control system architecture

In this section, we present the implementation of extended ABAC in XACML policy engine [10].

### A. ABAC system architecture

The system, in Figure 5, consists of the following modules.

1) User is a Man Machine Interface acting as the intermediate layer between system and physical person.
2) PEP handles request from user and forwards it to policy decision point for further policy evaluation.
3) Recourses are the digital assets that are securely stored in system storage or smart devices.
4) Obligation is a module handling different obligations that user or system needs to fulfil (e.g. notification).
5) PDP is responsible for validating the access control policy. It consists of three modules.
   a) Environmental attribute validation (EAV) is responsible for retrieving the environmental information from policy information point (PIP) or external information system.
   b) OSAV is responsible for validating the object's and subject's attributes. These attributes are generally retrieved from PIP.
   c) User validation is responsible for analysing user's request and his behaviour in order to determine if user is legitimate or malicious.
6) "Enforcement against compromised account" is responsible for enforcing some actions aiming to prevent malicious user from accessing devices. This process is complex and sometime requires a lengthy procedure that user or system needs to follow. For example, if system detects that user is suspicious, system may require user to prove his identity either by answer question (question and answer registered when creating account) or use other credential to prove.

### B. Implementation

In order to test our concept, we implement ABAC in XACML. Some scenarios in smart home are used for validation and testing. We also develop the behaviour analysis engine in Java and integrated with XACML engine.

**Testing data set.** In order to test our proposed security countermeasure approach, we need access history. We generate a simulated access history for 50000, 100000 and 1000000 transactions (records) stored in the access log file with the

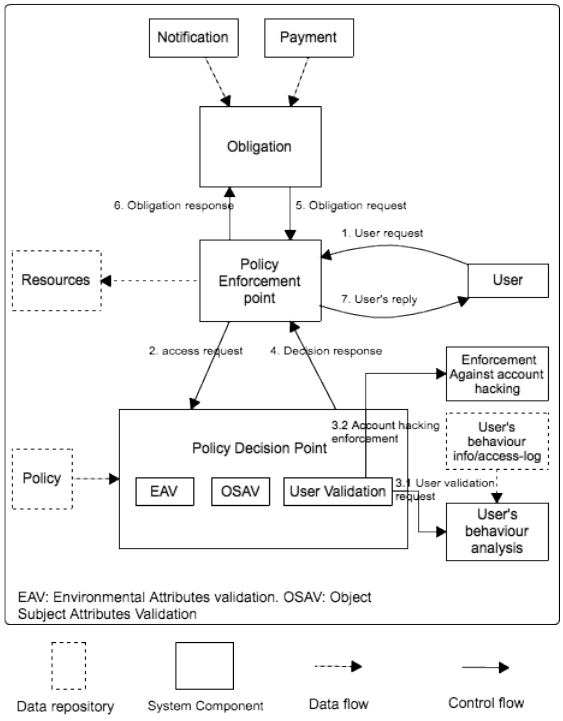


Figure 5. ABAC access control architecture supporting mechanism against compromised account

data structure like in Figure 3.

**Performance evaluation.** The idea is to evaluate the general performance of ABAC system with our security countermeasure approach. Since we use access history as the source of information for access pattern extraction and user's behaviour analysis, larger access history can introduce larger delay for access control policy evaluation. Thus, it is important to observe the policy evaluation processing time given different sizes of access history. We used 60 access control policies expressed in XACML policy language for testing. We simulated 50 different access requests and find the average policy evaluation processing time. We tested our system in Macbook air 1.3 Ghz Intel Core i5, memory 8 GB 1600 MHz DDR3. The result is shown in table I.

As expected, the policy evaluation processing time is increased in accordance with the size of the access log (see Table I). In case of load system, this issue can be a big challenge. However, there are two possible ways for reducing the policy evaluation time. The first option is to minimise the size of the access-log; another is to increase the computational power of the system (e.g. parallel computing).

To minimise the size of access-log, we need to minimise the size of observation interval. One solution is to divide a large observation internal into many smaller intervals (equal size). Then, we define the threshold (required maximum probability that user is legitimate) of each interval. The final

Table I
POLICY DECISION PROCESSING TIME

| Experiment No | Log size | Average processing time |
|---|---|---|
| 1 | 50000 records | 156 milliseconds |
| 2 | 100000 records | 203 milliseconds |
| 3 | 1000000 records | 580 milliseconds |

threshold value, which is used in policy, is an average of the threshold values from the smaller intervals. With this method, the size of access log used to calculate the value is the size of access log of one interval (the most recent access-log), not the entire access-log. For example, instead of using one year access-log, we can use a month access-log to evaluate the rule (policy).

We also tested the false negative (user requesting access is the real user, but the system permits with condition because of the absence of access pattern in the past) ratio to evaluate how the system reacts to abnormal behaviour of user. We tested in third experiment (see table I). The following information are used to generate access-log.
User = Edward, David, Jean, Pascal, Marie.
Action = turn-on, turn-off.
Device = CCTV, refrigerator, door, light.
Time = 6PM , 7PM , 5PM , 4PM, 10AM,11AM, 9AM.
The combination of user, action, device and time is selected randomly from above sets to form a record. In our experiment, the access pattern "David, turn-on, CCTV" is observed for different time interval and the behaviour analysis variable is set to 0.1443 (based on data set in access history). We created 4 different policies for David with 4 different behaviour analysis thresholds (b): b $\geq$ 0.1443 (policy 1), b $\geq$ 0.3 (policy 2), b $\geq$ 0.5 (policy 3), b $\geq$ 0.8 (policy 4). We tested 100 different requests and we find that if the threshold is set too high, most of requests are permitted with condition (see Section IV.B). This shows that user's access does not concentrate at a particular time. Among 100 requests (randomly pick) and when policy 1 is used, 8 requests return false negative. It is worth noting that this test is small, we need to test it with larger sample (number of users and access log). We consider this task as one of our future work.

## VI. RELATED WORK

Many efforts have been made in the research community to address the security issues in smart home and IoT system. Below are some researches relating to authorisation, access control and solutions to threats in smart home system.

N. Komninos et al [13] focuses on issues related to the security of the smart grid and the smart home, they present an integral part of the smart grid. Based on several scenarios, they present some of the most representative threats to the smart home/smart grid environment. The threats detected are categorised according to specific security goals set for the smart home/smart grid environment, and their impact on the overall system security is evaluated. However, this paper

does not address and implemented a specific solution for any threats. Authors presents a general threats from their defined scenarios and their overall security impacts on system.

M. Schiefer [11] defines the definition of smart home and categorises different products associated with smart home. The author also provides incidents and analyses in smart home context to estimate potential security threats in the future. The security threats in smart home are well studied by author, but there is not specific solutions to address them.

Rahul et al [2] proposed an access control model for home automation devices, which offers the capabilities to identify and connect physical devices into a unified secure system. The authors proposed to use Access Control List (ACL) as the access control security model to manage access to devices. Although ACL is simple to both understand and implement, it is not suitable for complex and fine-grained access control policies, which are needed in many IoT scenarios.

Bruce et al [1] conducted a security analysis and improvements of authentication and access control in the Internet of Things. The authors proposed the improvement protocol for authentication and access control by introducing the cryptographic key in both authentication and access control processes. RBAC is author's primary study in the paper. Authors also built their system to validate its performance and the result indicates that the improved protocol possesses many advantages against popular attack [5], and achieves better efficiency at low communication cost.

Ricardo et al [4] proposed a model-based security toolkit for IoT, which is integrated in a management framework for IoT devices, and supports specification and efficient evaluation of security policies to enable the protection of user's data. The authors's work is applied to a smart city scenario. The access control model, the authors used for building the frame work, is the improved RBAC model where the concept of trust and trust relationship is introduced.

## VII. Conclusion

In this paper, we propose the security countermeasure in case of compromised account and the general architecture of smart home system. Although the experiment shows that our solution provides promising result, there are many more work that needs be done. Especially, we need to test our system with larger sample and most importantly deploy this system in real home environment so that the access history sample can be collected and analysed. Our future work is to look at the device management and the secure policy administration toolkit for smart home.

## Acknowledgment

www.enmieux.be

FEDER
UNION EUROPEENNE
Wallonie
LE FONDS EUROPÉEN DE DÉVELOPPEMENT RÉGIONAL
ET LA WALLONIE INVESTISSENT DANS VOTRE AVENIR

## References

[1] Bruce Ndibanje, Hoon-Jae Lee and Sang-Gon Lee. Security Analysis and Improvement of Authentication and Access Control in the Internet of Things. Open access Sensors, 14(8), 14786-14805; doi:10.3390/s140814786, 2014.

[2] Rahul Godha, Sneh Prateek and Nikhita Kataria. Home Automation: Access Control for IoT Devices. International Journal of Scientific and Research Publication, Volume 4, Issue 10, October 2014, ISSN 2250-3153.

[3] Blase Ur, Jaeyeon Jung and Stuart Schechter. The current State of Access Control for Smart Devices in Homes. Workshop on Home Usable Privacy and Security (HUPS), July 24-26, 2013, Newcastle, UK.

[4] Ricardo Neisse, Gary Steri, Igor Nai Fovino and Gianmarco Baldini. SecKit: A Model-based Security Toolkit for the Internet of Things. The journal of Computer and Security (2015).

[5] Sachin Babar, Parikshit Mahalle, Antonietta Stango, Neeli Prasad and Ramjee Prasad. Proposed Security Model and Threat Taxonomy for the Internet of Things. International Conference on Network Security and Applications (CNSA 2010). Recent Trends in Network Security and Applications pp 420-429.

[6] J. Sathish Kular and Dhiren R. Patel. A survey on Internet of Things: Security and Privacy Issues. International Journal of Computer Applications. Volume 90. No 11, March 2014.

[7] Assessement of Access Control Systems. National Institute of Standards and Technology. Technology Administration U.S. Department of Commerce. http://csrc.nist.gov/publications/nistir/7316/NISTIR-7316.pdf

[8] Agrawal Rakesh, Imieliński Tomasz and Swami Arun. Mining Association Rules Between Sets of Items in Large Databases. SIGMOD Rec. June 1, 1993. Vol 22, No 2., NY, USA.

[9] Rath Th. Annanda and Colin Jean-Noël. Protecting Personal Data: Access Control for Privacy Preserving Perimeter Protection System. Data and Applications Security and Privacy XXIX: 29th Annual IFIP WG 11.3 Working Conference, DBSec 2015, pages="233–241, Fairfax, VA, USA, July 13-15, 2015.

[10] Extensible Markup Language (XACML). http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

[11] M. Schiefer. Smart Home Definition and Security Threats. Ninth International Conference on IT Security Incident Management IT Forensics. pages. 114-118, May 2015.

[12] Jose L. Hernandez-Ramos, Antonio J. Jara, Leandro Marin and Antonio F. Skarmeta. Distributed Capability-based Access Control for the Internet of Things. Journal of Internet Services and Information Security (JISIS), volume: 3, No: 3/4, pp. 1-16.

[13] N. Komninos, E. Philippou and A. Pitsillides. Survey in Smart Grid and Smart Home Security: Issues, Challenges and Countermeasures. IEEE Communications Surveys Tutorials. vol. 16, No. 4, P. 1933-1954, 2014.