July 2019

Assessing Vulnerabilities in IoT-based Ambient Assisted Living systems

Ioana-Domnina CRISTESCU [a,1] and José Ginés GIMÉNEZ MANUEL [b,2] and Juan Carlos AUGUSTO [b,3]

[a] *TAMIS Team*
INRIA, Rennes, France
[b] *Research Group on Development of Intelligent Environments*
Department of Computer Science, Middlesex University, London, UK

Abstract. Ambient Assisted Living systems aim at providing automated support to humans with special needs. Smart Homes equipped with Internet of Things infrastructure supporting the development of Ambient Intelligence which can look after humans is being widely investigated worldwide. As any IT based system, these have strengths and also weaknesses. One dimension of these systems developers want to strengthen is security, eliminating or at least reducing as much as possible potential threats. The motivation is clear, as these systems gather sensitive information about the health of an individual there is potential for harm if that information is accessed and used by the wrong person. This chapter starts by providing an analysis of stakeholders in this area. Then explains the IoT infrastructure used as a testbed for the main security analysis methods and tools. Finally it explains a process to assess the likelihood of certain vulnerabilities in the system. This process is mainly focused on the design stage of a system. It can be iteratively combined with development to inform a developing team which system architectures may be safer and worth given development priority.

Keywords. Ambient Assisted Living, Attack Tree, IoT Model Checking

1. Introduction

Ambient Assisted Living systems [1] have been developed for several years already. However being such a complex combination of technologies and having such a potential impact in humans lives, require extra care in design and development. One important advantage of such systems is that they provide an extra layer of care to people, especially when for circumstantial reasons better care is not available. So for example, older people prefer to live in their own independent space for as long as possible, however as they age more special care and precautions are needed and may be there are no other humans who can provide appropriate support at times. Systems which care raise alerts during emergencies are then useful. Also other, subtler, assistance is equally important. For

[1] Ioana-Domnina Cristescu. E-mail: ioana-domnina.cristescu@inria.fr
[2] José Ginés Giménez. E-mail: j.gimenezmanuel@mdx.ac.uk
[3] Juan Carlos Augusto. E-mail: J.Augusto@mdx.ac.uk

example, users starting to experience cognitive decline gradually start living out of synch with healthy life rhythms and some phenomena such as day-night misalignments and sun down syndrome can be observed in some cases [2,3].

Collecting fundamental life style patterns is useful to predict, advice, anticipate, and in some cases being able to react to emergencies saving time and reducing the negative effects of acute ill-health situations. Often the best person to assess the lifestyle information is outside the place where the information is gathered. For example, a person may be looked after by a smart home and those who need to have access to the system diagnosis on whether a change on medication led to better quality of life may be placed at a healthcare organization. Being able to securely transfer such sensitive information is an important part of the system. Current systems include sophisticated mechanisms to transport information securely from A to B, for example by using sophisticated security protocols including complex encryption mechanisms. The weakest links at this point in history are the participation of humans in the process (e.g., how do we know the person reading the information at a hospital is the one the data is intended for) and also the weak security mechanisms in various satellite technologies (e.g., Internet of Things gadgets).

This chapter explores the perception of users about the security of healthcare information being collected in domestic environments and transported to another environment for processing. First we show the results of questionnaires we run with various stakeholders. Then we explain the infrastructure which has been used to test a system prototype to aid the design of safer systems. Lastly a modelling system is illustrated showing how the possibilities of vulnerabilities can be assessed in a given system, in this case illustrated with the infrastructure previously described.

2. Stakeholders' Perceptions

As part of our research project we have routinely gathered stakeholders' perceptions, focusing on three main groups of them: system developers (S), healthcare professionals (H), and technology end users (U). We gathered opinions through an online questionnaire at various events and stakeholders workshops totalling 48 respondents (S:14, H:14, U:18, Others:2). Some questions were aimed at all respondents whilst other questions where aimed at specific stakeholder categories. There was a mix of multiple choice and open questions.

The following questions were addressed to end users only and from the 18 respondents in this category they clustered themselves as follows for each of the questions

"How much do you know of the internal working of the security mechanisms applied to your data?": 61% chose 'not enough', 17% chose 'enough' and 22% chose 'a considerable amount'.

"How much would you like to know of the internal working of the security mechanisms applied to your data?": 5% chose 'not much or nothing', 40% chose 'just enough' and 55% chose 'a considerable amount'.

"How much are you prepared to let the system know about yourself if that translates into greater security for your information?": 64% chose 'very little', 36% chose 'a lot'.

"What part of your private information would you be prepared to disclose if that would guarantee a better monitoring for your specific condition?": 33% chose 'Personal information: name, DOB, Phone number', 0% chose 'Current location and activity (use

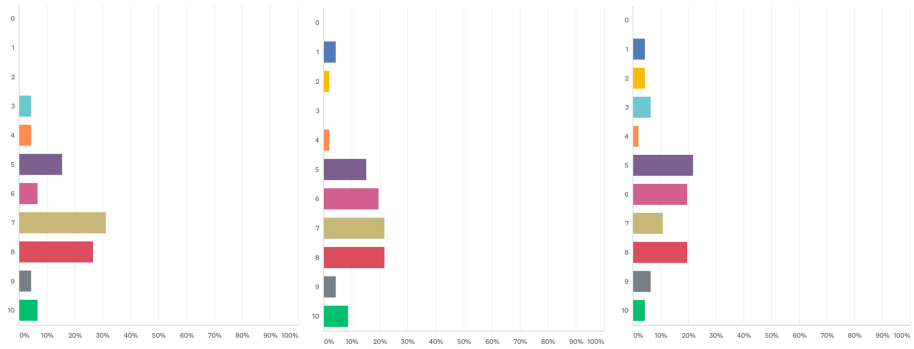
of electronic equipment, e.g. phone)' and 47% chose 'Anonymized medical data but including age, weight, height, etc.', and 20% chose 'None of the above unless I can see and understand how and where it is used'.

Amongst the questions aimed at all respondents we collected the following:

Question 12: "In a scale of 0-10 how high do you rate the security of the information you transfer through the tools which are most important to your specific work?"

Question 13: "In a scale of 0-10 how high do you rate the level of security provided by the tools which are most important to your specific work?"

Question 14: "In a scale of 0-10 how high do you rate the user-centred flexibility of the security mechanisms that these tools offer which are most important to your specific work?"



(a) Answers to Question 12. (b) Answers to Question 13. (c) Answers to Question 14.

Figure 1. Statistic extracted from the survey. The x-axis represents the percentage of participants who chose a value of proposed scale from 1 to 10 (Y-axis).

Some take away messages of the results above indicate the reluctance of users to share much personal information and the distrust on systems and tools handling their personal data as well as the unfriendliness and lack of transparency of the tools they rely on.

3. Pilot infrastructure

A pilot prototype was developed to create a real system wherein security concerns can be tested by different tools. The pilot deployment was carried out within the Smart Spaces lab at Middlesex University. Part of the Smart Spaces lab is set up as a smart room for experiments within IoT and for use of the Research GrOup On Development of Intelligent EnvironmentS. Figure 2 shows an accurate map of the lab with hardware elements installed inside as server and sensors used, smart hub and processing unit.

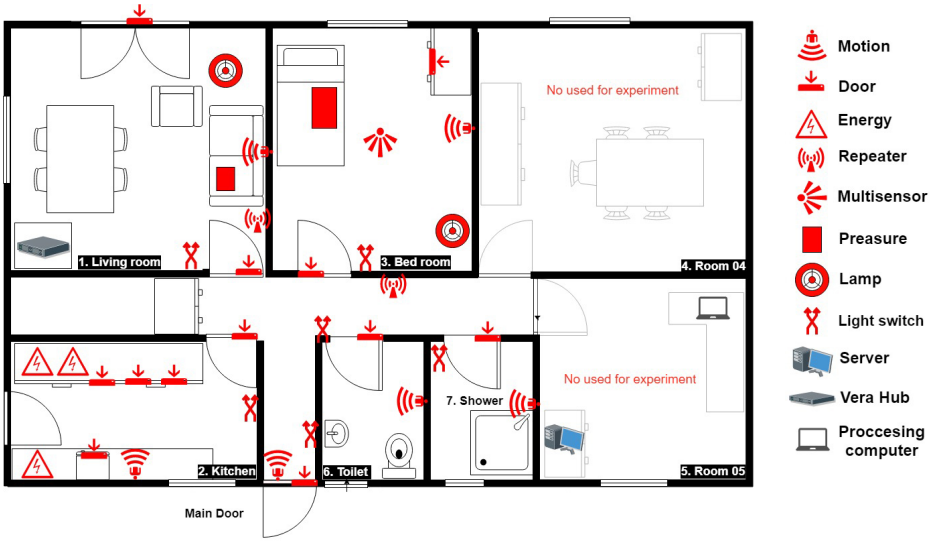


Figure 2. Lab map and hardware distribution.

3.1. System architecture

The approach to design the pilot architecture is based on create a smart home which manages sensitive user's information. This simulates a technological healthcare indoor environment wherein the security concerns can be audited. The Pilot background is based on indoor user's activity recognition focus on dementia such as [4] shows. In addiction, the user can provide personal health input through a mobile [5] such as blood rate using a smart watch. Figure 3 shows the Pilot architecture and next sections describe each element in more detail.

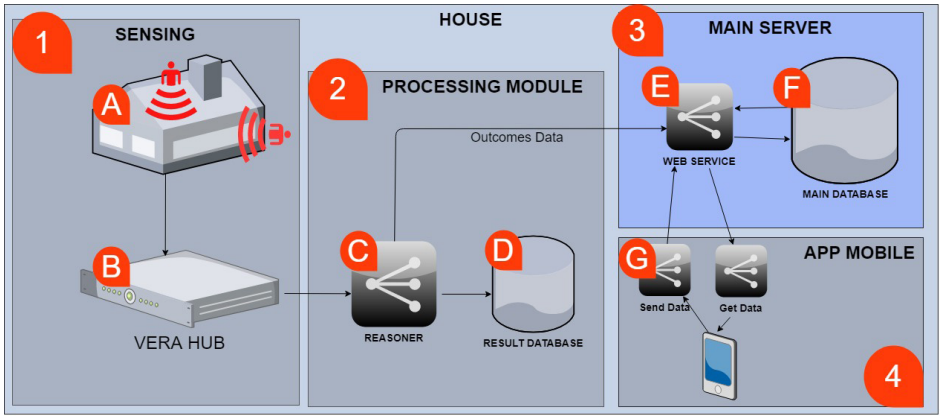


Figure 3. Initial Pilot 1 architecture

1 - The house sensing environment:

A - Z-wave sensing network.
B - Vera Hub.

2 - Processing module

C - MReasoner tool
D - MReasoner's database.

3 - Main Server

E - MySQL Database.
F - Web server and PHP API RESTful.

4 - Mobile Enviroment

G - Android mobile APP

3.1.1. Sensing environment

Sensing environment is the component which collects information from the house related to user's actions (e.g.: opening doors, switching on lights, etc.) or environment (e.g. temperature in a room, humidity or quantity of light) by using sensor devices. This element consists in a set of sensors distributed around the house and an smart hub which manages them. The smart hub installed is a Vera Plus model which uses its own wireless Z-Wave network to manage the sensors installed in the lab. The Z-Wave security implementation features 128-bit encryption. Vera does not use a database but it stores devices configuration and properties in JSON files and also writes in a non-persistent log the information from sensors, e.g. the change of state of devices. JSON files and log can be queried by external elements such as processor module (reasoner in figure 3) through 88 port by using HTTP protocol. The installed devices range from motion sensors which can detect movement in a place; switchers which informs about whether the light is on/off; energy sensors which can give information about the appliance plugged; pressure sensors placed on bed or chairs detect whether someone is sitting on it and reed sensors which are installed in doors, windows, cupboards, wardrobe and fridge door reporting if a door is open or close. Figure 2 shows a precise picture of sensors' location in the lab.

3.1.2. Processor module

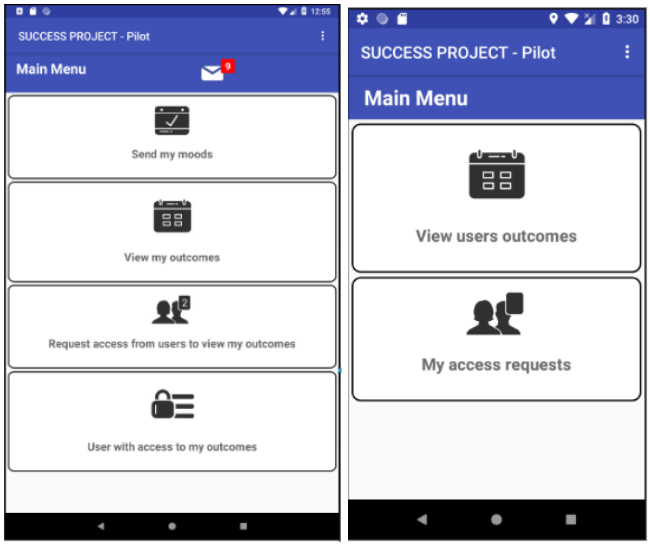
The Processing Module (PM) requests the sensing information collected by Vera. PM consists of temporal reasoning tool (MReasoner [6]) which infers sensors states and extracts logical conclusions about the user's context. To illustrate what represents user's context an example used in this project is the Activity Daily Living (ADL) recognition. These activities such are eating, sleeping, bathing, cooking or dressing are carrying out in the house by the user. PM can determine the activity being performing with some degree of verisimilitude. This activity recognition task is valuable in health environments such as in houses of people with cognitive decline or dementia wherein the primary user, the person living with dementia, is monitored. The gathered information about ADLs is related to when the activities occur and how long the user spends doing them. This information evidences behavioural patterns and deviations which can indicate an impairment in user's cognitive capacities. Beside professionals can use this information for user's evaluation, these systems provide efficient real-time monitoring which supports the caregivers in their supervision tasks over patients [4] allowing them to take action in critical situations such as user's falls.

3.1.3. Main server

The lab server (main server) plays the role of a normal server hosts in the cloud. It is in charge to store the user's health information to be accessible by doctors or other persons involved, as well as by the own user, from any place and device. The basic server configuration is similar to standard cloud servers. Thus, all connection with the server from external devices are through HTTPS protocol hence those connection using other protocols are refused. The web server manages a CRUD RESTful API developed in PHP. It provides the layer to retrieve information from MySQL database. The API implements register and login procedures using SHA1 password encryption, therefore, a user needs to authenticate in the system to reach sensible data. The API also manage sessions, cookies and other mechanisms related to security process such as blocking the user account, is user exits, after three login attempts which avoids brute-force attacks.

3.1.4. APP mobile

This component represents a direct input from users. They can connect to the cloud server sending and receiving data. The actions available are registration, log in, request and send information. The mobile interface displays different Graphical Interfaces (GUI) according to the user's role varying the available actions. Figure shows the main menu of both roles after the login process.



(a) Primary user main menu. (b) Secondary user main menu.

Figure 4. Mobile APP interface.

4. A case study

A common processes in client-server applications are registration and log in. These mechanisms play a crucial feature on security due to it grants access to users' information. The absence or deficiency of these procedures can report undesirable situations. In this sort of environment an unauthorized person can access to user's personal information and misusing it without the owner approval and unknown aim. We think it is important not just to analyse this process from the software point of view but also to show the users the risk associated with their behaviours and personal actions in the system.

The current pilot provides a registration system in the server allowing users to create an account based on their unique email and password chosen by them. A registered user is associated with a role which can be a primary user (PU) or a secondary user(SU). Depending on the assigned role a user will be able to access different information after log in. In this pilot the PU represents the person which is sending personal data to the server, either using the mobile or through the house (sensing environment). Thus, a logged PU can send data directly to the server and saved them but also can delete the account, delete data and give or withdraw grants to SU for access PU data. A SU represents doctors, caregivers or relatives interested in accessing PU's information. Initially, SU does not have access to any data. Thus, the first action available for a logged SU is to request authorization for access to PU's information. Once the PU permits SU access, SU can visualize user health information gathered whilst the PU does not revoke the access. Next section describes the transformation process from a probabilistic Pilot IoT model to SBIP system which evaluates the probability of an attack within the proposed context.

5. Modelling and evaluating a Pilot attack

Figure 5 is a graphical representation of the Pilot formal model described in the following sections. From this model we develop the security analysis based on the framework extensively described in [7].

5.1. Normal System, without an Attacker

The Primary User has two threads: a first one, called *sensorData*, sends signals to the sensors in the house. The second thread, called *giveAuth*, is used when a secondary user asks permission to access the primary user's data on the cloud.

$$\begin{aligned}
\text{enterRoom} &= PU \xrightarrow[\text{on}]{\text{sensors}} Eq \\
\text{sensing} &= PU \xrightarrow[\text{data}]{\text{sensors}} Eq \\
\text{exitRoom} &= PU \xrightarrow[\text{off}]{\text{sensors}} Eq \\
\text{sensorData} &= \text{enterRoom.sensing.exitRoom.sensorData} \qquad (1) \\
\text{giveAuth} &= PU \xleftarrow{\text{speaking}} SU.(PU \xrightarrow[\text{authPU}]{\text{https}} SU.\text{giveAuth} + \tau.\text{giveAuth}) \\
\textbf{PrimaryUser} &= \text{sensorData} + \text{giveAuth} \qquad (2)
\end{aligned}$$

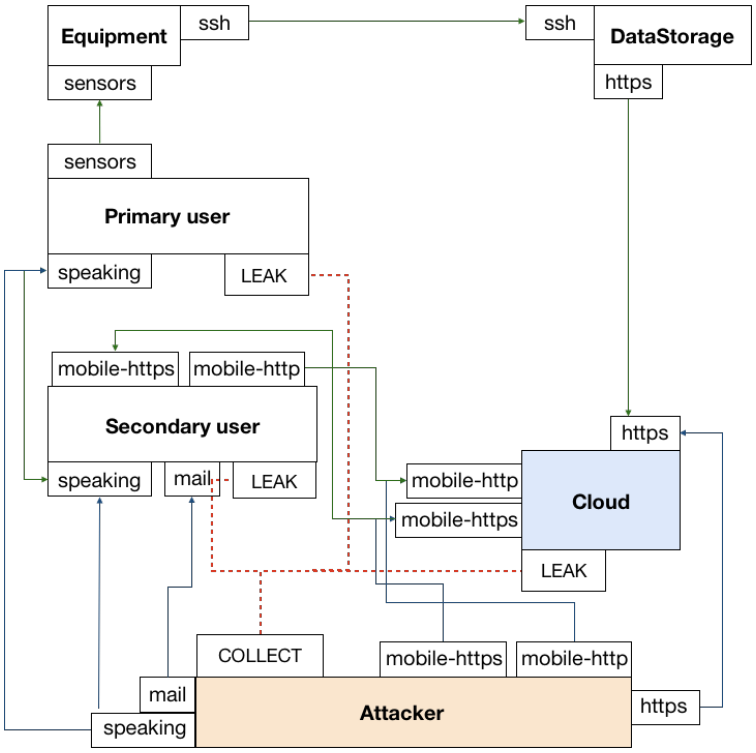


Figure 5. The SmartHome model. The green lines represent normal communications, the blue ones communications with the Attacker and the red dotted lines are leaks occurring in the system.

A Secondary User has to first to get an authorization from the Primary User to access its data. It can then access the data stored on the cloud. Note that the protocol used for accessing the Primary User's data requires an authorisation. Afterwards, either the user logs out, or it is logged out by the system after a timeout.

$$\begin{aligned}
\text{getAuthorisation} =& SU \xrightarrow[\text{getAuth}]{\text{speaking}} PU \\
\text{timeout} =& \tau \\
\text{queryCloud} =& SU \xrightarrow[\text{credentials}]{\text{mobile-https}} Cloud.SU \xleftarrow{\text{mobile-http}} Cloud. \\
& SU \xrightarrow[\text{info_request}]{\text{mobile-https}} Cloud.SU \xleftarrow{\text{mobile-https}} Cloud. \\
& (SU \xrightarrow[\text{logout}]{\text{mobile-https}} Cloud.\text{queryCloud} + \text{timeout.queryCloud})
\end{aligned} \tag{3}$$

$$\textbf{SecondaryUser} = \text{getAuthorisation.queryCloud} \tag{4}$$

The Equipment, consisting of the sensors in the house, are forwarding all data captured to the data storage unit.

$$\textbf{Equipment} = Eq \xleftarrow{\text{sensors}} PU.Eq \xrightarrow[\text{data}]{\text{ssh}} DS.\text{Equipment} \tag{5}$$

The Data Storage works as a server in the house. It receives and stores data, modeled by the action *receiveRawData*, and does some analysis on them to compile *behaviour logs*, which are then send on the cloud, using the action *sendBehaviourLog*.

$$\begin{aligned}
\text{receiveRawData} &= DS \xleftarrow{\text{ssh}} Eq \\
\text{sendBehaviourLog} &= DS \xrightarrow[\text{BehaviourLog}]{\text{https}} Cloud \\
\textbf{DataStorage} &= \text{receiveRawData.DataStorage} + \\
&\qquad \text{sendBehaviourLog.DataStorage}
\end{aligned} \tag{6}$$

The Cloud receives behaviour logs from the data storage, shown in *receiveBL*, and provides an api for querying the data stored, modeled by *queryAPI*.

$$\text{receiveBL} = Cloud \xleftarrow{\text{https}} DS \tag{7}$$

$$\begin{aligned}
\text{queryAPI} = {} & Cloud \xleftarrow{\text{mobile-https}} PU.Cloud \xrightarrow[\text{cookies}]{\text{mobile-https}} PU. \\
& Cloud \xleftarrow{\text{mobile-https}} PU.Cloud \xrightarrow[\text{info}]{\text{mobile-https}} PU. \\
& (Cloud \xleftarrow{\text{mobile-https}} PU.\text{queryAPI} + timeout.\text{queryAPI})
\end{aligned} \tag{8}$$

$$\textbf{Cloud} = \text{receiveBL.Cloud} + \text{queryAPI} \tag{9}$$

5.2. System with an Attacker

The Attacker has several lines of attack, modeled by the thread *AChoice* and shown in Figure 6. The attacker also collects leaks from the Primary and Secondary Users and from the Cloud.

$$\text{attackCloud} = A \xrightarrow[\text{getEmail}]{\text{https}} Cloud.\text{AChoice}$$

$$\text{attackSecondaryUser} = A \xrightarrow[\text{getEmail}]{\text{speaking}} SU.\text{AChoice}$$

$$\text{phishing} = A \xrightarrow[\text{getCredential}]{\text{mail}} SU.\text{AChoice}$$

$$\text{attackPrimaryUser} = A \xrightarrow[\text{getAuth}]{\text{speaking}} PU.\text{AChoice}$$

$$\text{getSensitiveData} = A \xrightarrow[\text{login}]{\text{mobile-https}} Cloud.A \xleftarrow{\text{mobile-https}} Cloud.$$
$$A \xrightarrow[\text{get\_sensitive\_info}]{\text{mobile-https}} Cloud.A \xleftarrow{\text{mobile-https}} Cloud.\text{AChoice}$$

$$\text{AChoice} = [a_1]\text{attackCloud} + [a_2]\text{attackSecondaryUser} + [a_3]\text{phishing} +$$
$$[a_4]\text{attackPrimaryUser} + [a_5]\text{getSensitiveData}$$

$$\text{collectPrimaryUser} = A \twoheadleftarrow PU.\text{collectPrimaryUser}$$

$$\text{collectSecondaryUser} = A \twoheadleftarrow SU.\text{collectSecondaryUser}$$

$$\text{collectCloud} = A \twoheadleftarrow Cloud.\text{collectCloud}$$

$$\textbf{Attacker} = \text{AChoice} \mid$$
$$\text{collectPrimaryUser} \mid \text{collectSecondaryUser} \mid \text{collectCloud}$$

The Primary User in Eq.2 has a new thread, *leakAuth*, where it gives its authorisation to an attacker, through a social attack.

$$\text{leakAuth} = PU \xleftarrow{\text{speaking}} A.([p_1]PU \xrightarrow[\text{authPU}]{} A.\text{giveAuth} + [p_2]\tau.\text{giveAuth})$$

$$\textbf{PrimaryUser} = \text{sensorData}(Eq.1) + \text{leakAuth}$$

The Secondary User in Eq.4 has two unsafe communications with the Attacker. It can choose between giving his email address, in the thread *leakEmail*. And it also has to choose between giving away his credentials or not, in the phishing attack modeled by *leakCredential*. For simplicity, we do not model the normal behaviour of the Secondary User when considering the attacks.

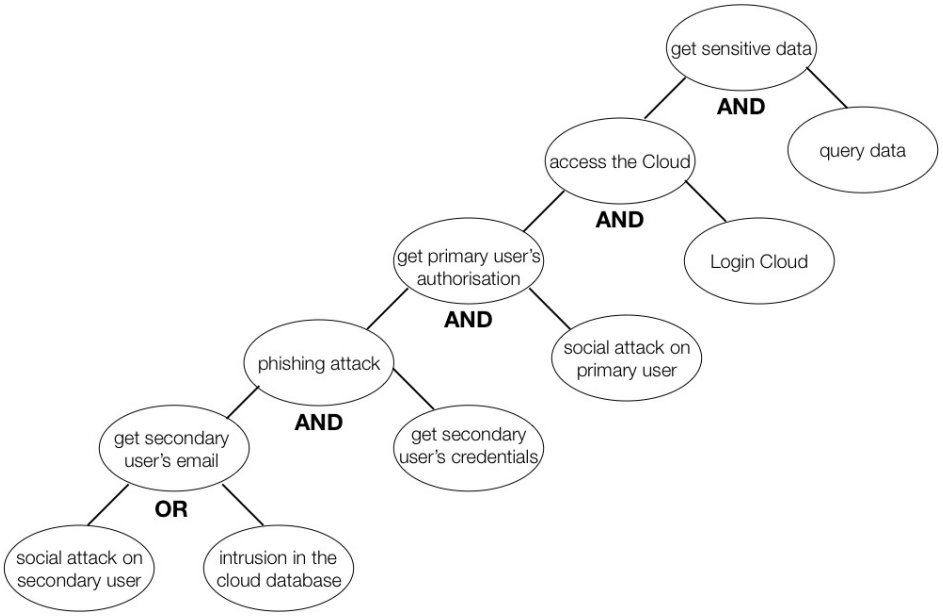


Figure 6. Attack Tree

$$\text{leakEmail} = SU \xleftarrow{\text{speaking}} A.([s_1]SU \xrightarrow[\text{emailSU}]{} A.\text{leakEmail} + [s_2]\tau.\text{leakEmail})$$

$$\text{leakCredential} = SU \xleftarrow{\text{mail}} A.$$
$$([s_3]SU \xrightarrow[\text{credentialSU}]{} A.\text{leakCredential} + [s_4]\tau.\text{leakCredential})$$

$$\textbf{SecondaryUser} = \text{leakEmail} + \text{leakCredential}$$

The Equipment in Eq.5 and Data Storage in Eq.6 have the same behaviours in the system under attack.

The Cloud in Eq.9 has the same behaviour as before, except that it can also leak (or not) the email of the Secondary User. Note that the thread *queryAPIAttacker* is similar to the one in Eq.8, but an *extra protection* step is added, to make the attack more difficult to succeed.

$$\text{queryAPIAttacker} = Cloud \xleftarrow{\text{mobile-https}} A.Cloud \xrightarrow[\text{cookies}]{\text{mobile-https}} A.$$
$$Cloud \xleftarrow{\text{mobile-https}} A.\text{extraProtection}$$

$$\text{extraProtection} = [c_3]Cloud \xrightarrow[\text{sensitive\_info}]{\text{mobile-https}} A.\text{Cloud} + [c_4]\tau.\text{Cloud}$$

$$\text{leakEmail} = Cloud \xleftarrow{\text{https}} A.$$
$$([c_1]Cloud \xrightarrow[\text{emailSU}]{} A.\text{leakEmail} + [c_2]\tau.\text{leakEmail})$$

$$\textbf{Cloud} = \text{receiveBL}(Eq.7) + \text{queryAPIAttacker} + \text{leakEmail}$$

	Monte Carlo		Importance Splitting	
Nb of Simulations	Result ($\times 10^{-4}$)	Time (s)	Result ($\times 10^{-4}$)	Time (s)
100	0	3,49	1,20	4,69
1000	0	5,12	1,43	8,17
10000	2,3	20,30	1,50	48,62
100000	1,2	399,82	out of memory	

Figure 7. Experiments.

5.3. Experiments

Figure 7 shows the results from running statistical model checking to estimate the probability of a successful attack. In the model we use for the experiments, a leak is five times less probable than an internal action.

From the figure we can infer the probability to be around 10^{-4}. We can see that Monte Carlo requires a larger number of simulations to estimate the probability of an attack, whereas importance splitting can eastimate the probability using fewer simulations, and in less time.

5.4. Other technical details

The protocols used are:

- *mobile-https* verifies users' authorisation; used in communication with the Cloud.
- *speaking* assumes physical proximity.
- *sensors* assumes physical proximity.
- *https* verifies that the right url is used to access the Cloud.
- *ssh* verifies that the data storage knows the equipment it receives data from.
- *mail* verifies that the attacker knows the email address of the user it tries to attack.

6. Conclusions

This chapter presents a practical example of using a framework to model, understand and analyse the security risks in a real IoT solution. Models aimed to real attacks analysis are useful as an immediate tool for identifying the attacks to a system as part of its security audit. However, we also consider these models as a good way of making security and privacy risks transparent to users, covering stakeholders concerns as expressed in the survey. Attack trees provide a bigger and clearer picture of situations that can jeopardize personal users information. This understandable information allows developing strategies addressed to stakeholders concerns by improving their knowledge about a system and security measures taken. Also, the final calculated probability of a successful attack in a system section where the human component is included offers a quantitative measure which is understandable by the general public (high risk/low risk). We are aware that the proposed Pilot is not the most secure solution and it probably has many security breaches to be improved. This is because its design has been constrained, just like in a real scenario, by the available resources like limited funds, time, among others. Nevertheless, using the proposed security analysis in early design stages can be beneficial to reach an

effective solution such as the case of study explained here. Although the chapter covers only one system attack scenario, the results show that there is a low attack probability in the proposed process. Thus, designers and developers can focus on analysing other procedures where risks could be higher. Hence, it can be said that this outcome allows them to allocate resources in other system modules where sensitive information is more exposed to potential harms. Thereby, the methods proposed in this chapter provide an understandable representation of the system risks that is useful for users and a quantitative analysis that is valuable for the developers.

References

[1] J. C. Augusto, M. Huch, A. Kameas, J. Maitland, P. J. McCullagh, J. Roberts, A. Sixsmith, and R. Wichert, eds., *Handbook of Ambient Assisted Living - Technology for Healthcare, Rehabilitation and Well-being*, vol. 11 of *Ambient Intelligence and Smart Environments*. IOS Press, 2012.

[2] P. McCullagh, W. Carswell, J. Augusto, S. Martin, M. Mulvenna, H. Zheng, H. Wang, J. Wallace, K. McSorley, B. Taylor, and W. Jeffers, "State of the art on night-time care of people with dementia," in *Proc. of the Conf. on Assisted Living 2009. IET, London*.

[3] N. Wolkove, O. Elkholy, M. Baltzam, and M. Palayew, "Sleep and aging: Sleep disorders commonly found in older people," vol. 176, no. 9, p. 12991304, 2007.

[4] I. Lazarou, A. Karakostas, T. G. Stavropoulos, T. Theodorosa, G. Meditskos, I. Kompatsiaris, and M. Tsolaki, "A novel and intelligent home monitoring system for care support of elders with cognitive impairment," vol. 54, no. 4, pp. 1561–1591, 2016.

[5] B. Reeder and A. David, "Health at hand: A systematic review of smart watch uses for health and wellness," *Journal of Biomedical Informatics*, vol. 63, pp. 269 – 276, 2016.

[6] U. A. Ibarra, J. C. Augusto, and A. A. Goenaga, "Temporal reasoning for intuitive specification of context-awareness," in *2014 International Conference on Intelligent Environments*, pp. 234–241, June 2014.

[7] D. Beaulaton, N. B. Said, I. Cristescu, and S. Sadou, "Security analysis of iot systems using attack trees," in *Graphical Models for Security* (M. Albanese, R. Horne, and C. P. Howar, eds.), Springer International Publishing, 2019.