Data Minimization in Distributed Networks for More Privacy

September 27, 2016

Abstract

The presence of connected devices in our environment is increasing. These devices form a network often called Internet of Things (or IoT for short), where everything from light-bulbs to thermostats can be controlled by an app or by another device. These services make a lot of that data available to the end user but also to malicious parties due to the devices leaking more data than intended or by bad design. This puts the end user at risk, violating its privacy and leaking sensitive data. One simple and obvious way to prevent leakages and misuses of personal data is to collect less of this data, a principle known as data minimization. However, this solution is rarely used in practice because of business models relying on personal data harvest on one hand and because of the difficulty to enforce it once it is defined what is actually needed to provide a service.

Acknowledgements

thanking everyone that's helped me throughout the course of the project

Contents

1 Introduction

1.1 Motivation

1.2 Aim

In this thesis I investigated ways to improve privacy in a special kind of IoT (Internet of Things) devices known as Wireless Sensor Networks (WSN). WSN are networks of autonomous sensors and actuators. The goal to enhance privacy for this kind of devices will be addressed by relying on data minimization. This means the project sought to improve privacy in distributed networks by limiting the amount of personal data being processed.

1.3 Scope and Limitations

prio writing this

1.4 Thesis Structure

saving for later when the thesis has shaped up

2 Background

This section should cover some background information to give the reader some background knowledge to what the project has been about that is required knowledge before moving forward.

2.1 Wireless Sensor Network (WSN)

A Wireless Sensor Network is recent improvement from the traditional sensor networks, made possible by advances in micro-electro-mechanical systems (MEMS) technology making sensor nodes that are smaller, multifunction and cheaper in comparison to previous sensors. Traditional sensors have two ways of being deployed; 1) They were positioned far away from the actual *phenomenon* (e.g. something known by sense perception) which required large sensors using complex techniques to distinguish the targets from surrounding noise. 2) Several sensors were deployed that only performed sensing and their communication topology had to be carefully engineered and they transmitted time series of the data to the central nodes which performed the communication.

Wireless Sensor Networks on the other is constructed by deploying a large number of sensor nodes close to the phenomenon and their position doesn't need to be engineered or predetermined.

references

discuss other good things with WSNs but try to keep it relevant

2.2 Data Minimization

As defined by the EDPS (European Data Protection Supervisor); "The principle of "data minimization" means that a data controller should limit the collection of personal information to what is directly relevant and necessary to accomplish a specified purpose. They should also retain the data only for as long as is necessary to fulfill that purpose. In other words, data controllers should collect only the personal data they really need, and should keep it only for as long as they need it."

argue why this quote is relevant

3 Theory

3.1 Basic WSN

As a starting point, a formulation of a basic Wireless Sensor Network was made so the ...

moar text

The basic Wireless Sensor Network consisted of a set of collection nodes (referred to as 'nodes'), a central server (referred to as 'the server') and finally an environment (the observed phenomenon).

can I use this wording here?

3.2 Formal Development

As a start for the formal development, the project required some 'building blocks' to better define certain aspects of a Wireless Sensor Network.

find a better word

From p. 127 in SPIN Ref Manual:

Definition 1. (FSA)
A *finite state automaton* is a tuple (S, s_0, L, T,F), where
S is a finite set of states,
s_0 is a distinguished initial state, $s_0 \in S$,
L is a finite set of *labels*,
T is a set of *transition*, $T \subseteq (S \times L \times S)$, *and*
F is a set of *final states*, $F \subseteq S$.

3.3 Model Checking

3.4 Promela & SPIN

Definitions 7.1-7.6 defined in Spin reference manual p.155-157

Definition 2. (Variable, Message, Message Channel, Process, Transition, System State)

adding all of these from book to define actors.

3.5 Related Work

3.5.1 Smart City

compare this to your work

3.5.2 Privacy Enhancing Technologies (PET)

compare this to your work

4 Specification

4.1 Definitions

4.1.1 Actors

Definition 1. (Actor)

An actor is a tuple (N, C, P), where

N is a *label* for referencing the actor,

a name basically?

C is a finite set of *message channels* and

P is a *process*.

Now to describe the interaction between two actors in the system, a behavior model was used:

< show image of a behavior model here >

Where the name of each actor is shown in the boxes at the top. The message channel used between them is shown as the arrows, where the arrow-head points

to the actor receiving the message and the message is referenced above it. Finally the ordering of the messages are in a descending order from the top, meaning the first message sent is shown furthest to the top of the figure and the horizontal lines at the end means the end of the communication.

noted?

can 'communication' be used?

4.1.2 Decisions

A decision, or a decision procedure is an algorithm that terminates with a yes or no answer given a decision problem.

ref 'Decision Procedure'-book

Definition 2. (borrow from 'Decision Procedure'-book, all relevant definitions are there)

4.1.3 Over-Collection

Definition 3. (Collecting)

A process P collects a data point d in a state s if after leaving state s (s_{+1}) then $d \in P.lvars$.

promela syntax for local variables, should be something else

Definition 4. (Over-Collection)

Over-collection is the state when a process collects more data than it requires to function.

Formal Definition: Let a process P be able to collect data entries and to evaluate boolean expressions.

$P_{eval} : D \rightarrow \textbf{Bool}$

Let a service $S(x, y, ...)$ be a boolean expression depending on variables $x, y, ...$

We say the process P dedicated to the service S, noted $< P, S >$, over-collects data if and only if P gets any data concerning one of the variables appearing in S after S has been evaluated to be true.

$< P, S >$ over-collects iff $\{D \in P_{collection}\} \wedge \{P_{eval}(D) = \textbf{True}\}$

something like that, but with S

4.2 Defining the models

As a starting point for defining the models, first different architectural choices were considered. This was done to help define different cases of Wireless Sensor Network using decisions. The different variations initially considered were:

- Centralized or Decentralized decision

- One or multiple sensor nodes
- Conjunctive or Disjunctive decision procedure

The first choice reflected how much the sensor nodes would analyze the data. Since Wireless Sensor Network has a processing unit, they could potentially analyze the collected data and make a decision on their own. The second choice simply reflected how many nodes were connected to the same server. The third choice reflected how the decision were processed, if the data from a single data point could trigger a decision or if the decision considered data from multiple entries before triggering.

'potentially' maybe irrelevant wording

entry?

To start of, models were made for each of the choices except conjunctive decision procedures. This was due to that a conjunctive decision procedure would require a more sophisticated algorithm to analyze the data than the other choices, which would require additional time for just one variation. Also this variation wasn't considered to be crucial to the project's aim, since the disjunctive decision procedure still presented interesting features for analyzing the system. So to start off, it wasn't focused on but still was kept as a consideration for further iterations.

promela, go?

5 Properties

5.1 Initial Model

The first model had properties for **safety correctness** and **liveness**. Due to the simplicity of the model, made both of them also rather simple to manage. The correctness property was stated as follows:

When over-collection has occurred (the decision is taken), the system should stop collecting.

In LTL: $\square(M \rightarrow (\diamond D)) \wedge (\diamond M)$

explain the last clause

Where **M** and **D** corresponds to the event that the message is sent and the collection is stopped respectively. Over-collection and the decision were rather interleaved in this system, the decision was taken at the same time the data was checked, therefore the brackets. The liveness property was stated as:

The program shall collect until over-collection has occurred.

In LTL: $\square(\neg D \text{ Until } (\diamond M)) \wedge (\diamond \neg D)$

until sign?

Where **D** and **M** are the same events as described previously.