EAI Endorsed Transactions
on Security and Safety

Research Article EAI.EU

SyntIoT: Privacy and security experimentation in consumer-oriented IoT ecosystems

Tomasz Kosiński[1,*], Riccardo Scandariato[1,2] and Morten Fjeld[1]

[1]Chalmers University of Technology
[2]Gothenburg University

Abstract

Since the advent of consumer-oriented IoT products, like smart homes, researchers have taken up the challenge of shielding the consumers from the risks this technology entails, including privacy harms. However, security and privacy research is 'hungry' for open data (e.g., about the network traffic patterns of the devices) and open platforms to validate IoT-related solutions outside a pure simulation environment. Except for the few cases seen in the related work, datasets are not readily available to the research community and are difficult to produce in-house. Also, the reproducibility of research results and open science is hindered by the lack of an open experimentation platform (to test privacy and security solutions) that also offers a fine-grained control of the experimental setup. We present SyntIoT, a platform that allows researchers to easily deploy a complete IoT ecosystem (including devices, users, vendor clouds) into the physical world and at a low cost, hence lowering the barriers to entry in this research field. SyntIoT can be used to collect field data and to realistically validate security and privacy solutions. Our platform uses synthetic IoT devices that are fully configurable in a declarative way. Interestingly, our platform also allows commercial devices to be deployed alongside the synthetic ones. The platform provides an infrastructure to monitor the ecosystem and to extract rich data, which can be used for empirical research and data mining. This paper presents the platform, explains how it meets established research needs not yet answered in previous works, and highlights its usage in the context of three experimental scenarios.

Keywords: IoT, privacy, security, open science.

Received on 03 October 2020, accepted on 14 November 2020, published on 16 November 2020

doi: 10.4108/eai.25-6-2021.170245

* Corresponding author. Email: tomasz.kosinski@chalmers.se

1 Introduction

Internet of Things (IoT) consumer devices such as video doorbells, voice assistants or smart light bulbs bring the vision of ubiquitous computing closer to fulfilment than ever before. The initial stage of wider adoption of consumeroriented IoT devices (i.e. those in focus of this paper) was centered around hubs, or gateways, capable of combining different devices to form a 'smart home' and automate the management of the users' household from the comfort of a smartphone. For instance, Samsung's SmartThings, was given special attention by researchers in the field of security and privacy [7]. The next wave of IoT adoption came with voice assistants that, differently from app-centred hubs, removed the need to use a smartphone in order to control devices like lights or garage door. These days, IoT devices are more and more integrated into larger IoT ecosystems that also include vendor clouds and third-party support services.

The convenience provided by IoT ecosystems has a price and it's often called privacy. Since the advent of the IoT paradigm, researchers have taken up the challenge of shielding the consumers of IoT technology from the risks it entails, including privacy harms. However, security and privacy research is 'hungry' for data and platforms to validate IoT-related countermeasures and solutions outside a pure simulation environment. An example of said countermeasures would be a permission system allowing end users of IoT to make informed decisions about the disclosure of the data their IoT devices collect.

In cyber-physical systems (like IoT), validation can be performed via a spectrum of experiments having an increasing degree of realism [13]. On one end of the

spectrum, we have 'in vitro' studies†, where the IoT ecosystem is simulated in software, sometimes with the possibility to have an actual device connected to the simulation rig as hardware in the loop, or HIL [9,18,5]. On the other end, we have 'in vivo' studies, where actual devices are deployed, connected to the networked ecosystem, and live data is collected. The two types of research studies do not exclude each other, as validation in the IoT domain naturally progresses from the former (in the proof-of-concept phase) to the latter (in the final prototype phase) [6].

This paper is positioned in the 'in vivo' camp, as we are interested in a solution that allows to collect live data about – and evaluate the efficacy of – actual privacy and security countermeasures in the cyber-physical space. In the literature, the two main 'in vivo' approaches to collect data and test privacy enhancing solutions are (i) lab studies, e.g., either small scale living labs where actual devices are installed in a space that is populated by users (typically, the researchers) or larger-scale open testbeds that federate several labs, and (ii) field studies enrolling volunteers that own IoT devices in order to monitor their spaces (e.g., homes) via software/hardware probes. As discussed in Section 2, most existing approaches do not allow to perform an experiment under strictly controlled and repeatable conditions. Some approaches are also difficult to both roll out and scale up, as there might be a high startup cost to acquire the devices, or the data gathering process might be slow due to the enrollment of volunteers. Further, the continuous evolution of commercial devices via firmware updates makes experimental results less reproducible.

As a scalable technical advancement, we propose SyntIoT, a new platform allowing researchers to effectively deploy a complete IoT ecosystem (including devices, users, vendor clouds) into the physical world. The platform (and the data it generates) is intended to be a realistic, in-vivo environment to prototype and test methods addressing privacy and security issues of consumer IoT ecosystems. The platform uses synthetic IoT devices, whose behaviour is fully configurable and controllable in a declarative way. As a benefit over other 'in vivo' approaches, SyntIoT allows the transparent access to the internals of the synthetic devices, e.g., their states, interactions with users, and so on. This introspection is particularly handy in some experiments, as discussed in Section 4. The platform also provides an infrastructure to monitor the ecosystem and extract measurements, which can be used for empirical research and data mining.

SyntIoT enables reproducible experimentation concerned with the understanding, controlling and compliance checking of the behaviour of consumer IoT devices. The SyntIoT experimental platform enables the collection of open data in a cost-effective way, lowers the barriers to entry for researchers, and supports open science by enabling reproducibility of research results. Furthermore, commercial IoT devices can be deployed alongside to the synthetic ones, for a rich testing and experimentation environment. As such, SyntIoT offers the possibility to train a technique on the synthetic devices and to test it on commercial ones, without changing the experimental setup.

This paper presents the SyntIoT platform in detail (Section 3) and explains how it meets established research needs not yet answered in previous works (Section 2.3). Making use of insights from development and experimentation with SyntIoT, the paper then presents three research-oriented scenarios (Section 4) that leverage SyntIoT to investigate the modelling of commercial devices, prototyping and testing of security and privacy countermeasures, and performing studies about usable security and privacy. Each of these scenarios demands answers to non-trivial needs from the area of security and privacy of consumer IoT. The platform is publicly available along with a sample dataset of collected open measurements [3].

2 Related Work

As mentioned in the introduction, this paper focuses on live experimentation of privacy and security solutions for consumer IoT ecosystems, as well as the 'in vivo' collection of the associated IoT data. In this context, privacy and security research has taken two distinct avenues. First, lab studies conduct measurements in IoT labs of various sizes and diversity, as described in Section 2.1. Second, field studies measure the activities of commercial devices at scale by deploying software (or hardware) at the homes of end-users, as discussed in Section 2.2.

2.1 Lab studies

IoT living labs. Several research groups work on solutions for IoT security and privacy problems with the use of self-established IoT Living labs. Apthorpe et al. [4] used a relatively small IoT lab setting to prototype and evaluate a countermeasure that avoids the inference of the user activities by a passive network observer that monitors the IoT traffic. Similarly, Acar et al. [1] used a lab setting including several IoT devices to demonstrate a machine-learning based attack on the privacy of IoT users allowing to identify in-home activities regardless of whether the traffic is encrypted or not. Then Ren et al. [15] performed measurements of a wide range of IoT devices in two geographically distributed laboratories which resulted in a comprehensive set of characteristics of information exposure of the users of these devices. The drawbacks of living labs are that (i) they are not open and (ii) entail a significant cost to set up. On the plus side,

† The term 'in vitro' is used to signify that the object of study (IoT devices, countermeasures, etc.) are observed outside their natural deployment context, which is the cyber-physical world.

they allow a tighter control of the experimental conditions.

Testbeds. IoT testbeds are open research platforms intended to facilitate large-scale, reproducible experimentation with IoT devices. Testbeds also allow the federation of smaller lab setups into larger networks and provide public APIs to access the platform and perform experiments. Examples of large-scale IoT testbeds are FIT IoT-LAB [2] and SmartSantander [16], as well as emerging services facilitating the aggregation of such testbeds, like FIESTA-IoT [17]. However, the main focus of existing testbeds is on the domains of wireless sensory networks, smart cities and cloud computing [6]. As such, testbeds do not offer consumer IoT devices present in users homes, which make the subject of privacy and security research pursued in IoT living labs. The advantage of testbeds is that they provide open and affordable access to existing infrastructure (sometimes at scale). However, the experimental setup is not under the control of the researchers performing experiments.

2.2 Field studies

Huang et al. [10] developed a data collection software released to volunteers who have been encouraged to run it in their home networks by offering the capability to observe characteristics of network traffic of their in-home IoT devices. At the same time, the volunteers contributed to a large dataset of anonymized descriptions of real-world IoT traffic informing and enabling further research towards protecting the volunteers themselves. Kumar et al. [11] applied another approach by partnering with an anvivirus provider who (with consent of the users) shared the results of user-initiated local network scans. This allowed for a large-scale analysis of the IoT ecosystem of end-users homes. Finally, Mazhar and Shafiq [12] studied IoT devices in the wild by using instrumented internet gateways deployed in over 200 homes and came up with a multidimensional characterization of observed IoT traffic.

2.3 Key features of SyntIoT

In Table 1, we offer a comparison of the approaches described above next to SyntIoT, so that the reader can weigh the pros and cons before making an informed decision of the most suitable approach for the research task at hand. In several dimensions, SyntIoT combines the strengths of lab studies and field studies. However, SyntIoT trades off the ability to produce datasets at scale (e.g., typical of testbeds) for a much finer-grained control of the experimentation set up. This is done in order to provide a unified experimentation platform for consumer IoT research, which enables reproducibility of results – in line with the open science principle.

Table 1. Based on a set of key features (rows), this table compares two existing ways to study IoT devices 'in vivo' (i.e., lab an field studies) to the way proposed in this paper.

	Lab studies (Section 2.1)	**Field studies** (Section 2.2)	**SyntIoT** (This work)
Open access	3(testbeds)	3	3
Reproducible results	7	3	3
Collection of datasets at-scale	3(testbeds)	3	7
Easy to achieve device variety	7	3[1]	3
Low cost	7(living labs)	7	3[2]
Control over the test environment	3(living labs)	7	3
Access to the internals of the observed devices	7	7	3[3]
Control over firmware versions	7	7	3[3]
Free of biased sampling [4]	7	7	3

[1] However, it is highly dependent on the current market adoption of the IoT devices within the demographic group available for the researchers to recruit from.

[2] There is no cost of shipping the hardware and no remuneration for study participants. Also, the time cost of recruitment and coordination is eliminated.

[3] Only for the synthetic devices.

[4] This results in a non-representative set of observed use patterns of devices owned by participants who might suffer from an over-representation of techsavvy users.

3 The experimentation platform

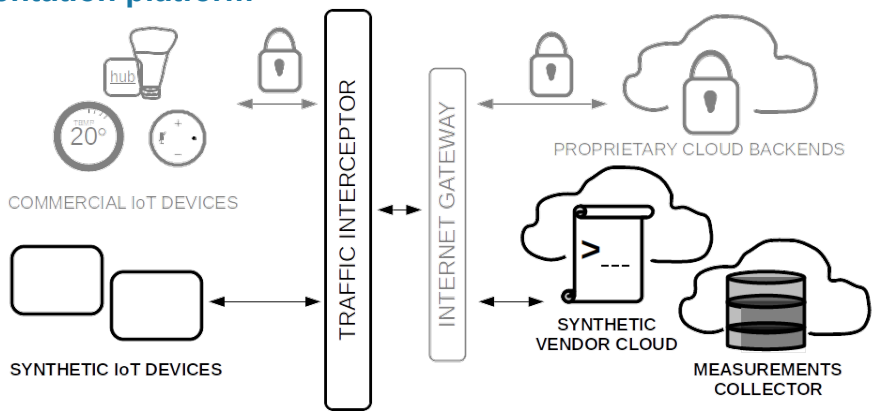

Fig.1. An overview of SyntIoT, the experimental platform introduced in this work. Components contributed by the authors (i.e. the synthetic IoT devices, vendor cloud, elements of traffic interceptor and the measurement collector) are emphasized with black colour. The platform can be deployed alongside proprietary, black-box IoT components (in gray in the figure).

The platform we present in this work consists of two primary parts, as depicted in Figure 1. The first part is the synthetic IoT ecosystem. It consists of fully transparent and controllable synthetic IoT devices and a synthetic IoT vendor cloud. The second part is the accompanying measurement infrastructure. It includes the interceptor and collector modules that process network traffic of observed IoT devices. Jointly, these two parts enable insights into the operations of the synthetic IoT devices across their entire operational life cycle. However, the platform enables studying not only synthetic but also commercial IoT devices within the same experimental environment.

The source code and configuration details allowing to deploy the platform can be found online [3].

3.1 SyntIoT: Synthetic IoT ecosystem

The synthetic IoT ecosystem consists of two main elements: the synthetic IoT devices and the synthetic vendor cloud. The former are intended to make it easy for researchers to deploy an IoT device characterized by a fully customizable behaviour. This enables emulation of several types of IoT devices in a lab setting while using only one common hardware platform capable of running the simulation software. The second component, the synthetic IoT vendor cloud, primarily supports the simulation of basic operations carried out by the synthetic IoT devices on the server side (e.g., phoning home, receiving updates and so on). Additionally, it offers an easy way to include, for instance, transparent mechanisms for server-side processing of the data collected from the devices within the platform.

Synthetic devices. The device emulation software can be executed on any GNU/Linux platform with a Python interpreter (the Python's package *transitions* [19] is required). For instance, the platform has been extensively tested on a Raspberry Pi. The behaviour of each synthetic IoT device can be specified in a declarative way as a set of parameters and a finite state machine (FSM) via a lightweight configuration language (as illustrated in Tables 2 and 3). It is worth mentioning that each device contains a reference to a synthetic IoT user defined in the respective part of the declarative specification. This definition describes the expected interactions with the synthetic IoT device.

FSM are one of the most widespread ways of specifying behaviour. However, defining a FSM could, in general, be a complex task. Nonetheless, IoT devices are usually single-purpose devices with an added internet connection and offer limited functionality. As such, they typically have a small number of states and, therefore, their specification as FSM is not cumbersome.

Declarative specification of device and user behaviour. The specification of the behaviour of synthetic devices is done via a file in JSON format. The configuration parameters that can be used in the specification can be found in Table 2. The specification is divided into three parts. The first part (Synthetic device section in Table 2) defines the parameters of the device, such as the device type, the volumes of data the device sends and receives, frequency of various connections and so on. For instance, a synthetic voice assistant device (which is typically streaming audio and might stream video content) will transfer higher amounts of data when compared to a light-bulb device. Devices of various types might also differ in the frequency of phone-home (heartbeat) connections, checks for updates and other characteristics that can be observed in their network traffic.

Table 2. Breakdown of the declarative specification of synthetic IoT devices and synthetic users. The three main entities to be specified are: characteristics of synthetic IoT devices, the FSM defining the behaviours of the synthetic device and behaviours of synthetic users. A subset of attributes was presented for brevity of the illustration.

Entity	Attributes	Exemplary values
Synthetic device	Type of device	Voice assistant, lightbulb hub, ...
	Max size of outgoing data flows [KB]	Voice assistant: 3072, lightbulb hub: 256, ...
	Heartbeat connections frequency [s]	Voice assistant: 50, lightbulb hub: 120, ...
	Firmware updates frequency [days]	Voice assistant: 21, lightbulb hub: 60, ...
Synthetic device FSM	List of states	off, booting up, online idle, offline idle, synchronizing time, ...
	Initial state	off
	FSM transitions	Powered up, done booting, detected user request, time sync triggered, ...
Synthetic user	Periods of activity	Night, early morning, lunchtime, ...
	Time ranges of the activity periods [h]	Night: [23.59, 6], Early morning: [6, 9], Lunchtime: [11.30, 13.30], ...
	Expected number of interactions with a synthetic device	(Voice assistant, Night, Weekday = Monday): (0, 2); (Lightbulb, Evening, Weekday = Monday): (0, 4), ...

Table 3. A list of states included in the default FSM model of a synthetic IoT device and the network traffic patterns associated with them.

State name	Associated network traffic patterns
off	no network traffic is observed
booting up	a burst of connections to a variety of remote entities can be observed
online idle	two types of distinct traffic can be observed. First, periodic time synchronization, heartbeat connections, API calls if a service such as IFTTT is being used and possible reporting of analytics. Secondly, irregular connections stemming from user interactions with the device or other events
offline idle	a sudden drop in connections in between online idle patterns characterizes this state, without a preceding booting up pattern
synchronizing time	contacting an NTP server
checking for updates	contacting a first / support party (CDN) in a regular fashion, periodically downloading a significantly bigger data package; the endpoint name might suggest its functionality
receiving user request	internal state of the device that might be challenging to observe though is necessary to simulate the way the device operates in real-world scenario; directly preceding service API calls
contacting first party	connection to the first party host, as determined based on observed traffic patterns and when not belonging to time synchronization or heartbeat patterns (the heartbeat connection is assumed not to interfere with normal operations of the device as a small and short-term event)
contacting support party / CDN	similar to contacting first party but when the provider hosts their service API using a Content Delivery Network (CDN) provider
contacting third party	contacting analytics and tracking services
responding to user request	internal state of the device that might be challenging to observe though is necessary to simulate the way the device operates in real-world scenario; directly following transfers of response information from the provider's API (hosted either in a first- or support party location)

The second part of the specification (Synthetic device FSM section in Table 2) contains the FSM definition. There, all states (including the initial state), transitions, conditions and desired behaviours need to be specified.

The researcher needs to provide handler methods responsible for the operationalization of (a) testing the conditions of the FSM transitions as well as (b) the behaviours attached to the transitions (e.g., the generation of network traffic). To increase the ease of use of the platform, we provide a default FSM definition including a set of defined states, transitions, conditions and behaviours of synthetic IoT devices. The pre-defined model is based on the knowledge and understanding of how commercial IoT devices work internally. This knowledge is derived from both the literature [15,10,8,1,4] and measurements conducted in our IoT lab featuring several commercial IoT devices. Table 3 provides an overview of default FSM's states along with the expected network behaviour associated with them. The researcher can easily customize the device by modifying the default FSM, or define their own FSM from scratch and override the default FSM specification. For instance, while adopting the default FSM to specific needs, one might choose to include connections to third party servers along with the associated network behaviours for a particular device. Similarly, the researcher might decide whether the service API of the synthetic device is hosted in the vendor's own cloud or with a support party / Content Delivery Network (CDN)‡.

The third part of the specification (Synthetic user section in Table 2) describes the behaviour of the user of the device, including activity periods and the frequency of device use within these periods. For example, a user might not interact with the device at night during weekdays but in early mornings the user might request news or weather reports, which would then trigger respective network activity of the synthetic voice assistant.

Implementation of the synthetic device and user behaviour. The translation of the declarative specification into the run-time behaviour of the synthetic device is handled with the use of the *transitions* Python package [19] for FSM orchestration. Additionally, synthetic IoT devices send out information of their internal state changes. This information includes: a timestamp, transition stage (beginning or finished), transition name, source state, destination state and a list of met (evaluating to true) and unmet (evaluating to false) conditions that are required for this transition to be activated. This information is intercepted by the traffic interceptor module (described in Section 3.2) and added to the measurements, as described later.

In Figure 2, we introduce an example of a complex device like a (synthetic) voice assistant. We remark that the figure is a simplification of the actual FSM of the synthetic device, in order to make the figure accessible to the reader. Thanks to the versatility of SyntIoT, this device and its interactions with users can be modeled with ease by configuring a few parameters and editing the FSM specification in a JSON file, but without having to code. The intended behaviour of the voice assistant (i.e., the FSM in Figure 2) and the network traces generated by SyntIoT are described in the following paragraphs. If the device is currently in the state "online idle" (a source state), the boot process has been completed, and a user request has been issued (conditions for the transition to be executed), the state machine of the synthetic device will execute a transition "done interpreting user request" to the destination state called "receiving user request". This transition will then trigger respective behaviour (defined as "update list of interactions with device"). Additionally, the transition will result in a set of subsequent state changes. First, the device will consume time intended to simulate interpretation of user's request. This is an equivalent of the local keyword recognition of voice assistant devices, i.e., when the device decides whether it has heard "Hey Alexa". Then, in state "contacting 1st party API", it will proceed to emit network behaviour involving connection to a first party hosting of voice assistant's service API. Specifically, it will transfer a certain amount of data that in real-world scenario typically comprises of the recording of user query, so that it can be interpreted in the vendor's cloud. Next, it will receive a certain amount of data in response and proceed to the state "responding to user request". Finally, it will consume time on responding to the user before returning to the online or offline idle state (depending on whether the internet connection is available at this point).

‡ We use the term "support party" after Ren et al. [15] to describe "any company providing [IoT device vendors with] outsourced computing resources, such as content delivery networks and cloud providers".

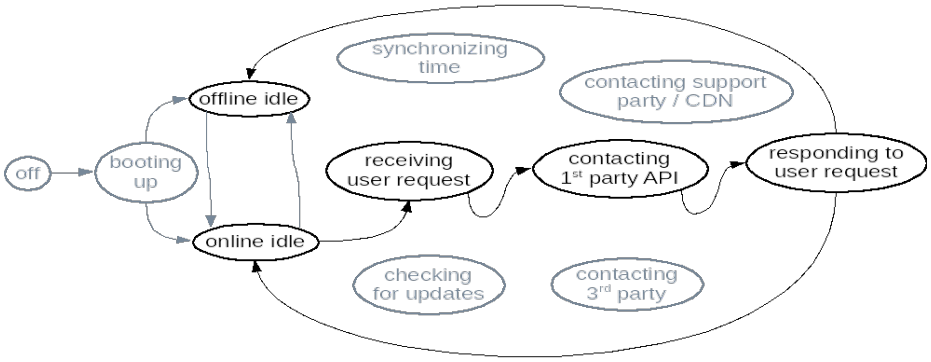

Fig.2. The finite state machine for a synthetic voice assistant. The black states are discussed in the paper. For readability, some states are grayed out and transitions are omitted.

Synthetic vendor cloud. The synthetic vendor cloud makes the second component of the synthetic IoT ecosystem of the platform. It is primarily aimed at supporting the basic operations of the synthetic device by acting as a simulated first/support party server hosting device's service API (e.g. voice services in the case of a synthetic voice assistant device). For the simulation of various remote cloud endpoints, multiple nodes should be used, each representing different kind of entities the synthetic device contacts. Typically these will include the first party server (vendor's cloud), optionally a support party server (e.g. CDN) and a third party (for analytics and tracking services). Additional kinds of cloud endpoints can be easily deployed using the synthetic cloud template provided within the platform. This basic template is based on the receive-and-respond functionality: it listens for requests from the synthetic device and returns phony data packages.

3.2 SyntIoT: The measurement infrastructure

To collect the network traces of the synthetic devices (as well as the commercial ones) we modified the *Princeton IoT Inspector*, an open-source software by Huang et al. [10]. The modified version has been tested on a macOS machine with a Python environment and the necessary Python dependencies. Measurements collected by the interceptor module are summarized in Table 4.

More precise measurements. The traffic information collected by SyntIoT for both commercial and synthetic devices is listed in the top part of Table 4.

Table 4. Collected information describing the traffic of IoT devices.

Data item	Brief description
Device identifier	Unique identifier of the IoT device, generated locally within the interceptor module
Packet timestamp	Timestamp [ms] indicating the moment of intercepting the packet sent or received by an IoT device
Window timestamp	Timestamp [ms] indicating the beginning of the current 5-second measurement time window (intercepted information is sent to the server every 5 seconds)
OUI	Identifier indicating the vendor of device's network chipset
DNS request flag	A flag marking DNS request packets

DSN requests and responses	Information extracted from DNS requests and responses sent and received by an IoT device (such as domain or resolver IP)
Device port	Port used by the IoT device to send or receive a packet
Remote port	Port used by the remote entity to send or receive a packet
Device IP	IP address of the IoT device used to send or receive a packet
Remote IP	IP address of the remote entity used to send or receive a packet
Bytes sent	Number of bytes sent per packet
Bytes received	Number of bytes received per packet
User Agent	HTTP User-Agent string
Hostname	Hostname used by the IoT device to request IP address from the DHCP server
	Uniquely for synthetic devices
State ID	Identifier of the state, for packets sent or received by the synthetic device while it remains in a state
Transition ID	Identifier of the transition, for packets sent or received by the synthetic device during a transition between states

The original IoT Inspector [10] (which we modified) is a network sniffer that continuously observes network traffic generated and received by the IoT devices. The IoT Inspector monitors the traffic for a time window of 5 seconds. If a device is active in that window, the IoT Ispector generates a log of the activity. The log includes an internally-generated device ID, the characteristics of the device itself (such as type and vendor), the contacted network destinations (or the sources that contacted the device), the total amount of bytes exchanged, the type of network encryption used, and so on. However, all these measurements are coarsely timestamped at the end time of each of observation windows (i.e., resolution of 5 seconds). We have modified the IoT Inspector to produce a better resolution, so that each network flow sent or received by a device is *accurately timestamped*, with microsecond precision.

Additional data for synthetic devices. Every time a synthetic device transitions to a new state of the FSM, it sends out a message with information about the beginning and end of the executed transition. This communication is then captured by the interceptor module. In this way, and as shown at the bottom of Table 4, all the measurements about the traffic of a synthetic device are associated to the device's internal state. This is particularly useful for data mining purposes. Note that the interceptor module intercepts also the traffic of commercial devices. In this case, however, no state information is tracked, since commercial devices do not share this information (obviously).

Forwarding to the collector. The data is regularly sent to the collector, which is shown in the right-hand side of Figure 1. This module is a server that receives the measurement data and stores it in a database. Note that the interceptor module only monitors those devices that have been explicitly specified by the researcher in the collector module, via a list that is retrieved before the interception (opt-in mechanism).

3.3 Limitations

First, the current implementation of SyntIoT offers a limited set of synthetic user behaviours (i.e. attempted interactions with the Simulated IoT device). It implicitly assumes an in-home scenario, while IoT devices are increasingly deployed outside of user's homes, e.g. in the hospitality industry [14].

Second, SyntIoT currently offers only one customizable model of the FSM simulating the behaviour of the IoT device. This model is based on educated guesses about the internal states and transition mechanisms of commercial IoT devices. However, the granularity of the states could be changed if knowledge of the internal behaviour of commercial IoT devices would become available by means of focused reverse-engineering.

We encourage the research community to contribute and share additional specifications of synthetic user behaviours and FSM models. Pull requests to the git repository are welcomed!

4 Experimentation scenarios

The platform introduced in this work provides functional, transparent and fully controllable abstractions representing the entire IoT pipeline, from devices to vendor clouds, and including the interaction with users. Below, we present a set of research scenarios (S1-S3) that can benefit from leveraging the SyntIoT platform in the research process. Following the description of each scenario we demonstrate how the intended use of the

platform will enrich the experimental toolbox of researchers. Each scenario is geared to research-oriented activities of: modelling commercial IoT devices (S1), evaluating countermeasures to activity inference attacks (S2) and testing permission systems (S3). These scenarios were chosen to answer non-trivial, documented needs in the practice of security and privacy investigation.

S1: Creating a ledger of synthetic twins for commercial IoT devices, hence enabling more reproducible and inclusive research. This research scenario is intended to construct a repository of synthetic IoT devices mimicking their commercial counterparts in terms of patterns of network traffic. This ledger can be used for reproducible research (as the synthetic devices are open) and can also foster inclusivity of researchers having no access to IoT labs or other experimental stacks otherwise (e.g. at the proof-of-concept stage of their work). SyntIoT provides a unified framework for iteratively collecting network traces of commercial devices, analyze them via machine learning to produce the configuration of the twin synthetic device, and measure the similarity of the network traces coming from the synthetic device vis-a-vis the commercial one. This process can be repeated over time in order to maintain the synthetic twins in sync with the commercial devices on the event of firmware updates. Currently, lab and field studies (as described in Sections 2.1 and 2.2) can provide datasets about the network characteristics of commercial IoT devices. These datasets are static, as they are collected under a fixed set of conditions in which the observed IoT devices happen to operate. As a consequence, updating or extending the knowledge provided by those studies is hardly possible. Scenario S1 and SyntIoT make it possible to transition from fragmented efforts dealing with the collection of datasets towards establishing open benchmarks for research on privacy and security in the domain of consumer IoT ecosystems.

S2: Supporting empirical evaluations of novel countermeasures to activity inference attacks on IoT users. This scenario leverages SyntIoT to further empirical research into privacy-preserving IoT traffic shaping algorithms, such as the work of Apthorpe et al. [4]. The privacy risks stemming from passive network observers inferring in-home activities of IoT users, even when the traffic is encrypted, involve physical stalking or planned burglaries. The risks increase with adversary's ability to observe targeted IoT users outside of the network traffic as well. As a result, IoT users living in shared housing of an attacker or when the attacker is an immediate neighbour are the most vulnerable.

Deployment of the state-of-the-art traffic shaping algorithms can be tested with the use of both synthetic IoT devices and vendor's cloud or just one side of the IoT ecosystem. This enables verification of their application feasibility beyond user-side devices such as home gateways or IoT hubs. Since the adoption of these privacy-preserving traffic shaping methods by less tech-savvy users or manufacturers of user-side devices will be a challenge, we see feasibility studies considering the use of IoT devices and clouds as an important step forward. Moreover, the user activity timing models used by the network traffic shaping algorithms can benefit from another feature of SyntIoT. That is, the capability of recording real user interaction patterns and subsequently storing them in the form of trained dynamic models would provide a necessary library of interaction patterns to create robust traffic shaping methods.

S3: Prototyping and testing usable privacy and security controls to empower the end users of IoT. This research scenario aims to develop usable and vendor-agnostic means of control over IoT devices for end users. The fragmented and proprietary IoT ecosystems created by individual vendors make it challenging for the end users to make informed decisions about not only the purchase of IoT devices but also subsequent disclosures of data the devices collect.

SyntIoT can support several aspects of this research track. First, the platform can be directly deployed and run within users' IoT environment (e.g. smart home) and the characteristics of network traffic of IoT devices could be sent to the platform's collector module. The existing end-to-end data collection modules (i.e. the Interceptor and Collector) can be augmented to support the collection of experience samples related to attitudes and expectations of the user regarding the IoT devices. Equipped with the insights into the technical and user perspective of the IoT environment operations, researchers could turn to addressing the main challenge. This is where the synthetic IoT ecosystem of the platform lends itself as an experimental framework for prototyping and testing a permission system developed on the basis of the identified privacy and security issues.

SyntIoT supports these experimental designs in several ways. To begin with, deploying the platform in the IoT-enabled environments of end users (typically homes) does not involve hardware cost (as in the case of laboratory setups, e.g. [15]), since the data collection software can run on commodity computers including those of users. At the same time, the entire data collection stack is open-sourced and made available to the research community to use, i.e. it can be run out of the box (as compared to tools that do not offer e.g. the cloud component of the experimental setup). Finally, by enabling simulation of both client and vendor ends of the IoT ecosystem, SyntIoT opens ways for devising functional prototypes of an IoT permission system unconstrained by the proprietary paradigms currently imposed by consumer IoT vendors.

5 Conclusion and future work

In this paper we have presented a platform that allows the deployment of a complete IoT ecosystem in a physical

space. The platform is open and allows the execution of research experiment under controlled and repeatable conditions. The platform is intended to facilitate and accelerate the research in the field of privacy and security of IoT systems. Through a set of three research scenarios (S1-S3), key uses of SyntIoT have been highlighted, with potential for additional research tracks.

We are adopting the platform ourselves for a number of research projects. Currently, we are testing a machine learning algorithm to extract the behavioural model of smart home devices from their network traces. In future work, we will include the human in the loop and present the extracted behaviour to the user in order to explain the privacy implications and receive feedback.

References

[1] Acar, A., Fereidooni, H., Abera, T., Sikder, A.K., Miettinen, M., Aksu, H., Conti, M., Sadeghi, A., Uluagac, A.S.: Peek-a-boo: I see your smart home activities, even encrypted! CoRR **abs/1808.02741** (2018), http://arxiv.org/abs/1808.02741

[2] Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., PissardGibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J., Watteyne, T.: Fit iotlab: A large scale open experimental iot testbed. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). pp. 459–464 (2015)

[3] Anonymized Authors: SyntIoT. https://codeberg.org/syntiot/syntiot (2020)

[4] Apthorpe, N., Huang, D.Y., Reisman, D., Narayanan, A., Feamster, N.: Keeping the smart home private with smart(er) traffic shaping. In: Proceedings on Privacy Enhancing Technologies. pp. 128–148 (2019), https://petsymposium.org/ 2019/files/papers/issue3/popets-2019-0040.pdf

[5] Bo¨hm, S., Kirsche, M.: Looking into hardware-in-the-loop coupling of omnet++ and rosenet (2015)

[6] Chernyshev, M., Baig, Z., Bello, O., Zeadally, S.: Internet of things (iot): Research, simulators, and testbeds. IEEE Internet of Things Journal **5**(3), 1637–1647 (2018)

[7] Fernandes, E., Jung, J., Prakash, A.: Security analysis of emerging smart home applications. In: 2016 IEEE symposium on security and privacy (SP). pp. 636– 654. IEEE (2016)

[8] Ford, M., Palmer, W.: Alexa, are you listening to me? an analysis of alexa voice service network traffic. Personal and Ubiquitous Computing **23**(1), 67–79 (Feb 2019). https://doi.org/10.1007/s00779-018-1174-x, https://doi.org/10.1007/ s00779-018-1174-x

[9] Gupta, H., Vahid Dastjerdi, A., Ghosh, S.K., Buyya, R.: ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. Software: Practice and Experience **47**(9), 1275–1296 (2017). https://doi.org/10.1002/spe.2509, https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2509

[10] Huang, D.Y., Apthorpe, N., Acar, G., Li, F., Feamster, N.: Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale (2019)

[11] Kumar, D., Shen, K., Case, B., Garg, D., Alperovich, G., Kuznetsov, D., Gupta, R., Durumeric, Z.: All things considered: An analysis of iot devices on home networks. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 1169– 1185. USENIX Association, Santa Clara, CA (Aug 2019), https://www.usenix.org/conference/usenixsecurity19/presentation/kumar-deepak

[12] Mazhar, M.H., Shafiq, Z.: Characterizing smart home iot traffic in the wild (2020)

[13] Papadopoulos, G., Gallais, A., Schreiner, G., Jou, E., Noel, T.: Thorough iot testbed characterization: From proof-of-concept to repeatable experimentations. Computer Networks **119** (2017)

[14] PwC Research: 2019 IoT Survey: Speed operations, strengthen relationships and drive what's next. https://www.pwc.com/us/en/services/consulting/technology/ emerging-technology/iot-pov.html (2019), last accessed 2020-04-01

[15] Ren, J., Dubois, D.J., Choffnes, D., Mandalari, A.M., Kolcun, R., Haddadi, H.: Information exposure from consumer iot devices: A multidimensional, networkinformed measurement approach. In: Proceedings of the Internet Measurement Conference. p. 267–279. IMC '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3355369.3355577, https://doi.org/10. 1145/3355369.3355577

[16] Sanchez, L., Mun˜oz, L., Galache, J.A., Sotres, P., Santana, J.R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., Pfisterer, D.: Smartsantander: Iot experimentation over a smart city testbed. Computer Networks **61**, 217 – 238 (2014). https://doi.org/https://doi.org/10.1016/j.bjp.2013.12.020, http://www.sciencedirect.com/science/article/pii/S1389128613004337, special issue on Future Internet Testbeds – Part I

[17] Serrano, M., Gyrard, A., Tragos, E., Nguyen, H.: Fiestaiot project: Federated interoperable semantic iot/cloud testbeds and applications. In: Companion Proceedings of the The Web Conference 2018. p. 425–426. WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). https://doi.org/10.1145/3184558.3186199, https://doi.org/10.1145/ 3184558.3186199

[18] Varga, A., Hornig, R.: An overview of the omnet++ simulation environment. In: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems Workshops. Simutools '08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL (2008)

[19] Yarkoni, T.: transitions: an object-oriented Python state machine implementation. https://github.com/pytransitions/transitions (2020), last accessed 2020-04-01