Towards Semantic Sensitive Feature Profiling of IoT Devices

Andrei Bytes, Sridhar Adepu and Jianying Zhou
Singapore University of Technology and Design
andrei_bytes@mymail.sutd.edu.sg, adepu_sridhar@mymail.sutd.edu.sg, jianying_zhou@sutd.edu.sg

Abstract—**Billions of Internet of Things (IoT) devices are being adopted in our daily life as personal wearables, home automation agents, medical appliances etc. Many domains of their use nowadays rely on the privacy and security of these devices - critical infrastructure, healthcare, logistics, manufacturing. In this paper, we aim to establish a standardized framework, which does not require access to physical devices yet allows to profile security and privacy-sensitive functionality in both existing and upcoming IoT products, based on semantic analysis of discovered technical information. We develop a software tool for automatic feature profiling of IoT devices and present case studies on two real-world IoT devices - a fitness tracker, Garmin Forerunner 230, and a voice-controlled home assistant, Amazon Echo Dot 2^{nd} Generation and further provide comparative results analysis.**

Index Terms—**Internet-of-Things, Security, Attack surface, Privacy, Capability profiling.**

I. Introduction

"The Internet of Things (IoT), [22] is a network of machines and devices capable of interacting with each other". Low-cost, energy-efficient and with broad capabilities [30], IoT devices are often powerful enough to execute software, ported from desktop machines and have extensive capabilities for data sensing and wireless network interconnection. Thus, Ronen et al. [12] presented a powerful city-scale attack on Phillips Hue smart lamps. The use of personal fitness trackers by US army soldiers[1] resulted in online leakage of sensitive army base location information. Many other incidents with IoT demonstrate the importance of applied research on security and privacy of these products.

A number of research works [25], [15] are focused on understanding security and privacy threats in IoT. However, much less research has been made on effective identification of actual sensitive capabilities in these devices. Hence, early security analysis is crucial due to the rapid development of hardware startups, metamorphosis of existing devices and entire product lines into IoT. The ability to perform security-sensitive capability profiling on such devices based on discovered technical information and without physical access to actual hardware is essential. A particular solution we are focusing on is the use of direct and indirect resources to extract information to identify security-sensitive capabilities of these devices. Building on semantic representation of the aggregated data, we apply relevant criteria sets to automatically reveal and profile such functionality. The proposed mechanism considers results obtained by static analysis of companion mobile applications of IoT devices; discovers vendor documentation and user manuals; crawls online information, certification registries and other available sources. To extend coverage to upcoming products, which are not present on the market yet, we propose the use of IoT startup feeds and crowdfunded device prototypes to estimate their technical characteristics at early stage. Potential consumers can use the proposed profiling methodology for security analysis of selected IoT devices. Additionally, this methodology facilitates multi-domain, large-scale security research in the areas of Enterprise, Medical, Industrial environments and Critical Infrastructures. This allows for the continuous discovery of vulnerabilities introduced by IoT components in these systems.

Contributions: 1) We propose a set of direct and indirect information discovery sources which do not rely on having physical access to devices to gather technical capability information. 2) We propose sample criteria sets for sensitive capability profiling, summarizing factors that enabled or amplified major IoT incidents highlighted in related works. 3) We implement the proposed approach as a modular software tool, which uses a sample generic criteria set. 4) We apply the presented methodology on two real-world IoT products to evaluate information availability in direct and indirect sources and to obtain capability profiles of these devices.

Remainder of the paper is organized as follows: In Section II, we present the context of the problem and capability profiling overview. We describe the feature discovery and profiling steps of IoT devices in Section III and Section IV. In Section V, we present a prototype of an automated tool for capability profiling of IoT devices. In Section VI, we conduct two case studies on Garmin forerunner and Amazon echo dot 2nd generation. Related work is discussed in Section VII, and the conclusion is given in Section VIII.

II. Problem Context, Capability Profiling

The number of vulnerability reports for widely used software and hardware IoT components is growing every day. User manuals, documents, firmware updates and other sources reveal crucial information to scope the devices and their new modes of use. This information can be used to prepare attack vectors, which were not known before. This situation exists for most modern IoT products used in many domains of our life. Most IoT devices nowadays are technically unified and have great similarity in their implementation stacks. Thus it is useful to analyze them in groups, based on a certain criteria, such as possible domains of application, form-factor, portability, the presence of common electronic components from the same vendors.

Part of this work is supported by SUTD start-up grant SRG-ISTD-2017-124

[1] https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases

Feature profiling overview Fig. 1 shows the processing steps involved in mapping privacy profiles from discovered information about an IoT device. The majority this information can be collected from publicly available means and further processed semantically.

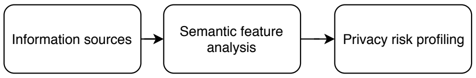


Fig. 1: Overview of the capability profiling framework

In this paper, the semantic approach is referred to as processing the input information which is then transformed into a single textual input feed in order to identify the mappable signs. These in turn point to security-sensitive device capabilities to be considered as a part of its threat surface. This data can be used to form device capability profiles and compose subsets of devices which are exposed to similar risks. The entire process can be divided into two major processing stages:

1) **Discovery** Collection of technical information about IoT devices into a unified feed. This results in output data about device technical capabilities which can be further passed to the profiling stage.
2) **Profiling** Based on device capability information obtained from the discovery stage in unified form, the feature signatures are mapped into a single privacy profile of the device.

III. Discovery stage

Automated analysis of input sources is performed in order to gather information about on-board functionality, technical components and general capability of the device with respect to sensitive data processing and modes of operation (sound recording, video capture, types of communication, transmission range, types of sensing and actuating equipment).

Direct information sources: published in clear form and freely available product specifications, descriptions, update changelogs, vendor-specific web resources, search engines cache, vendor-originated documentation and operation manuals.

Indirect information sources: information revealed by reverse engineering IoT device firmware updates, published source code of its components, automated static and dynamic analysis of control applications (companion applications for mobile, desktop platforms), specifications of particular hardware components, country-specific online registries for certification and regulation compliance.

Crawling direct information sources:

Searching for specifications: A specification collection module browses websites associated with IoT device vendors, as well as the websites of known OEM and peripheral brands for IoT products. Technical specifications of IoT products, published user manuals and other device-specific documents are collected into a local storage.

Extracting firmware releases and changelogs: The updates tracking module uses data feeds containing firmware release information to track changelogs, bugfixes and feature updates of IoT devices.

Monitoring search index: The web search engine bridge module queries DuckDuckGo[2] search engine to obtain recent mentions of upcoming IoT products, their concept presentations and advertised features.

Scraping articles and crowd funding platforms: The upcoming product crawler module discovers new IoT projects on their way to the market using top crowdfunding platforms[3] (Kickstarter, Indiegogo), blogs[4] (IOT Council, IOT.do and others).

IoT-specific search engines: Search engines like Shodan, Censys.io[5] serve as a source of network service information and fingerprints of hardware devices.

Processing indirect information sources:

Firmware analysis: Device firmware analysis is a crucial part of product functionality identification, as well as privacy capability and domains of use. Usually distributed in separate versioned update packages, it contains IoT device control logic compiled into binary files, filesystem structures, configuration files and hardware drivers. There are different approaches that can be used to obtain the firmware files [9]: download from vendor resource, capture during device update and direct extraction from hardware.

Control application analysis: The module searches for mobile applications which interact with a given IoT device and extracts the permissions and feature-specific functionality calls. Due to extensive market coverage by Google Android OS the module implements the analysis of APK packages. Thus, the fast mode (Algorithm: 1) gives maximum performance - it extracts Android Manifest file from the application package, decodes it into ASCII parseable form **A** and uses the preconfigured wordlist **w** to find the occurences of the capability tags. The found capabilities are extracted into a separate list **F** which is the output of the program.

The slow mode (Algorithm: 2) requires much more processing time and not only decodes Android Package Kit **x** into Android Manifest file **E** but also decodes the Dalvik bytecode parts into parseable .smali files **DF** and performs their deep analysis one by one for all occurrences of capability names from the preconfigured wordlist **w**.

Algorithm 1 Android APK analysis - Quick mode

```
Input: Android Package Kit x
Output: Found capability keys F
Ensure: Wordlist w
  F = []                      ▷ Found capability and declared permission keys list
  A = []                      ▷ ASCII representation of Android Manifest
  E = extract_xml(x)
  A = decode(E)
  i = 0
  while !EOF(w) do
      IF $w[i] \subset A$ then
              $F = F \cup w[i]$
      END IF
      i = i+1
  end while
```

Certification documents: Some countries require IoT products on the way to their markets to go through wireless communication testing and certification procedures. Knowing the product certification ID often makes it possible to obtain extensive technical details about general hardware used in the product

[2] https://duckduckgo.com

[3] https://www.kickstarter.com, https://www.indiegogo.com

[4] https://www.theinternetofthings.eu/, https://iot.do/

[5] https://www.shodan.io,https://censys.io

Algorithm 2 Android APK analysis - Extended mode

```
Input: Android Package Kit x
Output: Found capability keys F
Ensure: Wordlist w
  F = []                  ▷ Found capability and declared permission keys list
  DF = []                 ▷ Dalvik bytecode files
  O = []                  ▷ ASCII representation of Android Manifest and decompiled .smali instructions
  E, DF = extract(x)
  $P_{man}$ = decode(E)
  $P_{sm}$ = decompile(DF)
  O = O ∪ $P_{man}$ ∪ $P_{sm}$
  i = 0
  while !EOF(w) do
      IF w[i] ⊂ O then
              F = F ∪ w[i]
      END IF
      i = i+1
  end while
```

implementation, as well as its communication capabilities (wireless adapters, their power and transmission range). In particular, useful certification identifiers are those related to the following registers: FCC, ISACA, CMIIT, KCC MSIP, ANATEL.

Offline documentation parsing and manual contribution: The module receives as input offline documentation, such as PDF whitepapers and technical reports. The documents are parsed and semantically analyzed to identify features, components and their implementation context. Custom information can be contributed by users, such as appending known FCC ID of the hardware, which can usually be found on its case. A Wiki-like semantic user interface is used to categorize manual contributions for automated processing.

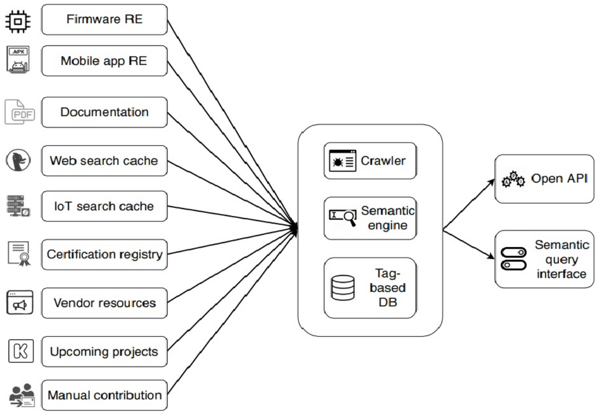


Fig. 2: Data sources

IV. Profiling stage

Information gathered at the previous stage is used as input for building capability profiles of IoT devices.

Building a criteria set A preliminary stage for the proposed semantic approach is the formation of relevant criteria sets covering security-sensitive functionality. While an average device consumer can rely on a combined criteria set, research teams can define additional criteria sets to correspond to the applied aims of their research (e.g. limit capability identification to target risks solely in medical or industrial domain). Building on analysis of major IoT security and privacy incidents from the relevant work listed in Section VII, we compose a sample combined criteria to reveal the presence of functionality which was previously identified as a factor for sensitive data leakage and attack propagation in popular IoT devices.

A proposed sample set is implemented in three separate criteria groups. The **Privacy-loss (PL)** criteria as shown in Table I describes hardware and software features and processed data classes which have potential impact on privacy loss caused by the device. A number of factors and implementation technologies relevant to confidentiality threats are grouped into **Confidentiality-implementation (CI)** criteria set (Table II). Finally, the **Impact-scale (IS)** criteria, described in Table III , contains properties related to the actual scale of a potential incident and the affected userbase.

Similarly, more advanced criteria sets can be compiled to fit more advanced and domain-specific research aims: 1) IIoT (Industrial IoT): focus on capabilities with high environmental and safety impact, dangerous connectivity of physical components. 2) Healthcare: focus on presence of biometric sensing, components with access to private data, medical records. 3) Consumer privacy: focus on use of microphone, camera, presence of tracking capabilities. 4) Network appliance: focus on signs of known vulnerable software, commonly exploitable implementation flaws.

The following capabilities of IoT devices are grouped under *Privacy-loss (PL) criteria*:

TABLE I: Privacy-loss (PL) criteria

	PL criteria	Components	Tag
1	**Sensing, recording capability**	Microphones	MI
		Cameras	CM
		Motion	MO
		Location	LO
		Biomedical	BM
2	**Primary usage domain**	Industrial (critical infrasct-turcture)	IF
		Industrial (manufacturing)	FC
		Healthcare	MD
		Enterprise	EE
		Consumer	CO
3	**Types of processed data**	Personally identifiable	PI
		Commercial	CO
		Anonymized only	AN
4	**Reuse modes and multi-user sharing**	Individual use	ID
		Disposable	DD
		Multi-user sharing	WH
		Multi-user (reinitialization and reuse)	WR

- *Sensing and recording capability* covers use of on-board hardware and software components with the capability to access potentially personal data. Thus, recording capability of the device can be represented by signs of use of microphone and camera access in the device business logic.
- *Primary usage domain* indicates market targeting of the device and its potential userbase. This information contributes to device privacy risk assurance.
- *Types of processed data* in context of privacy leak incidents are used as estimation of potential loss.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2903739, IEEE Internet of Things Journal

- *Reuse modes and multi-user device sharing* serve as risk indicators of multi-user data leaks with consideration to attack surfaces, formed by reuse of local device storage and using shared accounts at the remote backend.

Confidentiality-implementation (CI) criteria: The factors grouped under Confidentiality-implementation (CI) criteria reflect particular system implementation aspects which are to be considered as part of its attack surface and within risk estimation.

- *Access mode* of the device indicates the accessibility of the device in the context of exploitation of its attack surface.
- *Infrastructure points of trust* lists the components in the device infrastructure which are involved in the personal data processing flow.
- *Connectivity* ways supported by the IoT device hardware serve as traditional indicators to determine its attack vectors.
- *Wireless transmission range* is referred to as an indicator of discoverability of the IoT device threat surface and its potential role in further attack distribution to other devices.

TABLE II: Confidentiality-implementation (CI) criteria

	CI criteria	Components	Tag
1	**Access mode**	For public placement	AP
		For individual use	AI
2	**Infrastructure points of trust**	Data is handled by a host app (Android, IOS, PC apps)	AA,IO,PA
		Exposed to remote servers and services	RS
3	**Connectivity**	**Wireless interface**	
		BLE. Bluetooth 3	B4,B3
		WIFI	WF
		RFID	RI
		Zigbee	ZB
		GSM (2G)	2G
		3G, LTE	3G, 4G
		Proprietary (custom) radio protocol	RF
		Wired	
		USB	UB
		Ethernet	EH
4	**Wireless transmission range**	Short-range (Zigbee, BLE, RFID)	SR
		Medium-range (WIFI)	MM
		Long-range (Celluar, LoRa)	LN

Impact-scale (IS) criteria: Impact-scale (IS) criteria contains chosen properties of the IoT product and its posture on the market which have impact in the context of consequence scale caused by a potential privacy leak incident.

- *Product line maturity* is a product property considered as the potential of device hardware and its infrastructure to contain security flaws which cause privacy leaks. Due to the usual practice of model derivation from base models previously designed by the vendor, the maturity of the entire product line is considered for this criteria.
- *Market spread* refers to the current device userbase and means of product integration into lifestyle and business processes as a factor of potential incident impact scale.
- *Network operation topology* is considered as a factor of the device capability for attack distribution. It was previously shown [12] that IoT devices with built-in hardware support of peer-to-peer communications are exposed to additional risk of being used as a relay for large-scale attack distribution in a stealthy manner.

TABLE III: Impact-scale (IS) criteria

	IS criteria	Components	Tag
1	**Product line maturity**	Prototype or early revision	EM
		Stable revision	SM
		Market-mature product line	MM
2	**Market spread factor**	Specialized solution with limited consumer base	M1
		Average market spread, mixed domains of use	M2
		Consumer daily usage, large user base worldwide	M3
3	**Network operation topology**	Peer-to-peer	2P
		Host-only	HO
		Star	ST
		Mesh	MH

V. Software prototype: Automated Tool

Automation of the proposed IoT discovery and profiling approach enables the aggregating IoT devices capability information in a scalable way and enables categorization, further querying and machine processing of collected profiles. To facilitate this process, we develop a tool prototype written in Python with a modular structure described below. The modules use a predefined criteria set in the form of many-to-one dictionary structure of textual tags corresponding to device capabilities and return the presence of these tags in the inbound sources.

Documentation parser module: Analyzes textual representation of product specification documents, user manuals to extract the capabilities of a given IoT product and returns found functionality tags for profiling. For PDF files, the module uses PdfFileReader from PyPDF2 python package to obtain the textual representation before further processing.

Firmware analyzer module: The module relies on classic *BSD file*[6] and *binwalk*[7] utilities to determine the package format and extract the contents. After extraction, the printable ASCII data is used to semantically identify sensitive device functionality calls and capabilities against the used criteria set.

Android companion application analyzer module: The module is implemented in Bash and receives Android Package Kit (.apk) file as an input. A control flag is passed to the module to select between the Quick/Extended algorithm to use. In software implementation of the *Quick mode* algorithm, *AAPT* (Android Asset Packaging Tool) is used to extract the information from Android Manifest in the fastest possible manner. The API access permissions are decoded and extracted in textual form and compared against a criteria set. All obtained resources, SMALI classes, JSON files, XML and other ASCII files are saved in the same location and further processed

[6] https://www.freebsd.org/cgi/man.cgi?query=file&sektion=1

[7] https://github.com/ReFirmLabs/binwalk

in a single feed for finding and exporting the capability tag occurrences.

Shodan.io lookup module The module is implemented in Python and relies on *shodan*[8] package and is to be configured with Shodan API access key. The module executes a search query through Shodan API, returning data from the index for a given IoT device model which is analyzed to compare with the criteria set.

Certification discovery and parsing module The module uses csv parser[9] with Python 2.7 to process ISACA[10] certification lists to map the found information about device transmission capabilities into the corresponding functionality tags.

Control interface The software prototype provides a simple command line interface for the user, allowing her to start information gathering and profiling with minimal possible input. Thus, in its simplest form, the use case is as follows: a potential consumer (end user), before buying the IoT product she is interested in, passes a single -u <product_summary_url > parameter to the tool, pointing the crawler to a webpage which contains a description of the IoT product (the minimal requirement: vendor and model name). At the second step, the user may be asked to confirm the actual product version (model revision, sensors set etc) if this information cannot be extracted unambiguously by building solely on the page contents. The product information is then passed to discovery modules and after full processing, results in a capability profile of the single product. In this case of minimal configuration, direct and indirect information sources for discovery stage are selected automatically. Finally, the user can repeat the first step to pre-configure more information about the product to improve the results and to process similar products for comparison.

Another use case is oriented for large-scale device discovery and analysis. A control REST API shall be implemented for advanced interaction with the tool. This allows for working with large lists of products in a single batch, extention of targeted criteria sets, export of resulting profiles and monitoring their differences in real time, as well as extraction of device subsets by querying profile attributes.

VI. Case Study

This section presents two case studies: Garmin Forerunner 230 and Amazon Echo Dot 2^{nd} generation.

A. Garmin Forerunner 230: a running watch

Smart watches and fitness trackers are amongst the most widely used consumer IoT devices today. Normally, they can be interconnected with surrounding electronic appliances (mobile phones, desktop computers, vehicles, wireless beacons, heart rate monitors, running tracks) through built-in network interfaces. These products actively store, process and transmit the personal data of their users, such as geographical locations, physiological measurements, audio call activity and textual messages, pictures, visual recordings. Being a single wearable device from the user‘s prospective, they are rather complex IoT systems with interconnected infrastructure behind the scenes.

Garmin Forerunner 230‘s data flow affects multiple system components. A built-in GPS receiver chip (MediaTek MT3333) handles geographical coordinates and time adjustment. Nordic Semiconductor nRF51422 System-on-a-Chip empowers the watch with a wireless interface for Bluetooth Low Energy (BLE) communication. This allows the watch to connect to mobile devices, desktop machines, vehicles and many other devices.

The mentioned nRF51422 SoC also provides ANT+ wireless interface, which is used by Garmin to connect its sensors and accessories to the device. This chip supports multiple protocols and can be also used for custom data transmission in 2.4 GHZ radio band. The USB interface allows access to device flash memory (settings, location history and user fitness data as FIT files).

The watch has multiple companion applications to be controlled and synchronized from mobile and desktop operating systems. Thus, Garmin Connect, Garmin Express and connect.garmin.com web application are different applications and can trigger different capabilities of the watch.

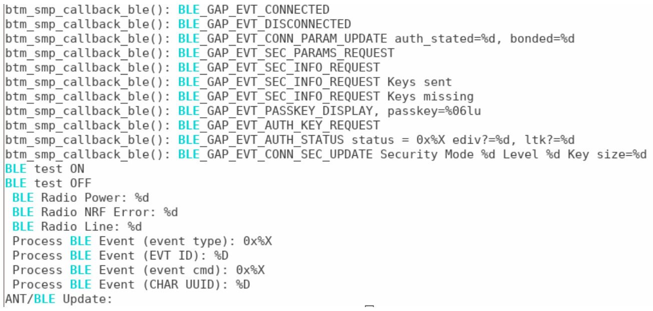
1) Direct information sources:

Technical specifications: User documentation for Garmin Forerunner 230 has been found on the Garmin website[11]. The document analyzer module was used to locate the functionality signatures that occurred in the owners manual.

Firmware release changelogs: The official Garmin Forerunner 235 support page [6], as well as its unofficial mirrors [5], contains the detailed release information for published firmware update packages and lists the changelog with mentions of new implemented features and deployed fixes in relation to previously existing device capabilities. Relevant functionality tags were extracted to a separate dictionary, based on the changelog records.

2) Indirect information sources:

Firmware analysis: The Garmin Forerunner 230 firmware package (GCD) is downloaded from Garmin website support section. Basic automated processing of the binary shows ASCII-printable strings which reveal the functionality related to Bluetooth Low Energy (BLE) communication capability (Fig. 3). The “nordic” keyword is detected in the file path (as



```
btm_smp_callback_ble(): BLE_GAP_EVT_CONNECTED
btm_smp_callback_ble(): BLE_GAP_EVT_DISCONNECTED
btm_smp_callback_ble(): BLE_GAP_EVT_CONN_PARAM_UPDATE auth_stated=%d, bonded=%d
btm_smp_callback_ble(): BLE_GAP_EVT_SEC_PARAMS_REQUEST
btm_smp_callback_ble(): BLE_GAP_EVT_SEC_INFO_REQUEST
btm_smp_callback_ble(): BLE_GAP_EVT_SEC_INFO_REQUEST Keys sent
btm_smp_callback_ble(): BLE_GAP_EVT_SEC_INFO_REQUEST Keys missing
btm_smp_callback_ble(): BLE_GAP_EVT_PASSKEY_DISPLAY, passkey=%06lu
btm_smp_callback_ble(): BLE_GAP_EVT_AUTH_KEY_REQUEST
btm_smp_callback_ble(): BLE_GAP_EVT_AUTH_STATUS status = 0x%X ediv?=%d, ltk?=%d
btm_smp_callback_ble(): BLE_GAP_EVT_CONN_SEC_UPDATE Security Mode %d Level %d Key size=%d
BLE test ON
BLE test OFF
 BLE Radio Power: %d
 BLE Radio NRF Error: %d
 BLE Radio Line: %d
 Process BLE Event (event type): 0x%X
 Process BLE Event (EVT ID): %D
 Process BLE Event (event cmd): 0x%X
 Process BLE Event (CHAR UUID): %D
ANT/BLE Update:
```

Fig. 3: Bluetooth Low Energy capability indication in the Garmin Forerunner 230 firmware

[8] https://shodan.readthedocs.io/en/latest/

[9] https://docs.python.org/2/library/csv.html

[10] https://www.isaca.org/pages/default.aspx

[11] http://www.garmin.com.sg/minisite/forerunner/

shown in Fig. 4), likely a trace of the source code of utility to work with Nordic Semiconductor System-on-a-Chip.

```
..\..\..\BTM\nordic\btm_utl.c: 82
```

Fig. 4: Bluetooth Low Energy capability indication in the Garmin Forerunner 230 firmware

Android companion application analysis Garmin Connect is published in Google Play store. The deep search method (involving application Dalvik virtual machine bytecode .dex files) was used.
Certification lookup The lookup of FCC ID IPH-A2758 of Garmin Forerunner 230 returns a set of device internal pictures and documents submitted to The US Federal Communications Commission [3]. The device hardware components can be identified from the picture of the product‘s internals; the wireless transmission power and frequency range are revealed.

Resulting device capability profile:

{MO,LO,BM}{MD,CM}{PI}{ID} ;
{AI}{AA,IO,RS}{B4,UB}{SR} ;
{MM}{M3}{2P,ST}

Conclusion: The discovery stage was facilitated by wide availability of device firmware packages of different versions. The obfuscation of the package was not made by the vendor, which helped to reveal the known hardware components used in the device design. Another important source of technical details, Android companion application, improved the coverage of analysis mainly from the list of required permissions and names of remote API method calls. The product market properties were assigned by manual contribution as the keyset did not cover the automation of this criteria. The profile indicates device individual use mode, as well as presence of extensive sensing and wireless communication capabilities.

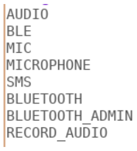
B. *Amazon Echo Dot 2^{nd} Gen: a smart home assistant*

Amazon Echo Dot v2 is capable of voice recognition, making calls, sending messages, downloading online content, music streaming, ordering goods and services. The specification also describes rich extension functionality to control other IoT devices: *“Controls lights, fans, TVs, switches, thermostats, garage doors, sprinklers, locks, and more with compatible connected devices from WeMo, Philips Hue, Sony, Samsung SmartThings, Nest, and others”* [1]. Echo Dot’s hardware is powerful enough to run heavy Android-based firmware and a wide set of Java applications on top of it.

Security and privacy Amazon Echo Dot v2, being one of the most affordable in Amazon Alexa - powered device line, is commonly placed nowadays in private households, hotels and office spaces. Being a powerful enough Android-based computation platform, the device itself is a potential target for hijacking, turning it into a remotely controlled in-network attack node. Equipped with seven far-field directed microphones, the device is capable of high-sensitive in-room sound recording and its transmission of the network to the remote Amazon Alexa backend. Implementation flaws in the device software and hardware stack or undocumented remote invocation of its features can cause direct security privacy threats in the context of sensitive personal and commercial data leakage.

1) *Direct information sources:*

Technical specifications A user manual has been extracted from FCC certification documents [2]. The document contains information pointing to the use of BLE, 802.11 (WIFI), Bluetooth 3.0 for wireless interconnection. The tags were located based on a wordlist using a semantic analysis module.
Firmware release changelogs The Amazon Device Support website section has been found [12], which contains the basic firmware release information for Amazon Echo product line. However, this source does not contain references to the changes of functionality in each release which could be used as input for further profiling.
Firmware analysis: At the moment of writing the firmware for Amazon Echo Dot 2 Generation was not freely distributed online. However, a direct link to firmware binary was previously extracted from the unencrypted network traffic during the firmware update process [7]. Analysis of the firmware update package shows that it is heavy binary, the device runs on Android-based Amazon Fire OS.
Android companion application analysis: Amazon Echo Dot architecture involves a companion app. We have scanned the Android version of Amazon Alexa companion app available online with our prototype semantic capability profiler. Semantic profiling based on our default wordlist reveals usage of microphone access and recording capabilities and Bluetooth 3.0 communication. Fig. 5 shows the output obtained using the relevant software module of our prototype.



```
AUDIO
BLE
MIC
MICROPHONE
SMS
BLUETOOTH
BLUETOOTH_ADMIN
RECORD_AUDIO
```

Fig. 5: Amazon Alexa companion app feature profiling

Certification lookup: Amazon Echo Dot has multiple revisions, certified and assigned with different FCC IDs. It was found that FCC ID 2AHSE-2045 is relevant to 2^{nd} Generation of the device. The US Federal Communications Commission database lookup reveals information about device internals and communication capabilities.

Resulting device capability profile:

{MI}{CM}{PI}{MH};
{AP}{RS}{B3,WF,UB}{MM}; {SM,M3,HO}

Conclusion: Amazon did not explicitly list the firmware packages on the website - an alternative download source was used. The US certification documentation was extensive and contained user manuals for multiple revisions of the device. The Android application is generic for the Alexa model line, however it contains functionality logic, specific to the researched device. The most capable source of technical information was represented by user documentation of the

[12] https://www.amazon.com/gp/help/customer/display.html?nodeId=201602210

device. Similar to Case Study 1, the market parameters had to be assigned as a manual information contribution.

C. Outcomes and comparison of results

1) The unified, programmatically processable mapping of the models of IoT devices to their sensitive capabilities are stored in the single database with a query interface. This gives an opportunity to perform mass queries to arrange the devices into particular groups (subsets) based on the research criteria.

   Example subset 1: A subset of devices generated by criteria $(\{WR, MO\} \cup \{MD, AI\} \cup \{EM\})$, contains all identified *early prototypes* of *medical* IoT products which are capable of *motion sensing* and are built for *individual use* and are intended to be *re-used* by the next owner.

   Example subset 2: Similarly, a lookup applying the criteria $(\{B4, EE\} \cup \{AP\})$ will generate a subset of *enterprise* IoT devices built for *placement in public areas* which are able to communicate via *Bluetooth Low Energy*, such as indoor employee tracking Bluetooth beacons [8].

   Example subset 3: Another criteria, $(\{IF\} \cup \{2G, EH\} \cup \{MM, M1\})$ will generate a subset of *market-mature IIoT (industrial)* IoT devices used by *critical infrastructure* which can be controlled by *Ethernet* or 2^{nd} generation GSM cell network.
2) As opposed to traditional IoT device discovery methods, such as Shodan.io [13] lookup, the proposed methodology is based on the semantic processing of the specifications and other information sources which gives an opportunity to explore not only IoT devices which are exposed to the Internet, but also information about *offline devices* and the devices that are in their early stages and are entering the market.
3) Lastly, in single-device mode the proposed methodology implementation in a form of software tool provides an instrument which can gather preliminary information and describes the attack surface of the device without actual analysis of the device.

D. Discussion and limitations:

Approach perspective for domain-specific IoT research parties The proposed approach for profiling IoT devices offers the opportunity to be extended through additional criteria sets to perform research driven by specialized domains of IoT device application, such as Critical Infrastructure, Healthcare, Household privacy and Enterprise confidentiality. Depending on device capabilities that are crucial for the particular research domain, research parties are encouraged to configure the framework with specialized criteria. Building product subsets based on mutual properties and the presence of common security and privacy-sensitive components enables researchers to further analyze devices which are affected by a certain attack, predict common implementation flaws and perform targeted vulnerability discovery.

Approach perspective for IoT device consumers Before purchasing an IoT device, a consumer can use the proposed capability profiling tool to gather information on what the device is capable of and thus what personal information can be leaked from the device in case of a security incident. A beneficial property of the proposed methodology is that it can be used for upcoming new devices. The user does not need to purchase the actual device (which might be not yet available for sale) to explore its capability implications. She can get a capability posture of it with automatically collected and profiled data.

Methodology limitations and future perspective: One of the major limitations of the discovery stage is the fact that not all devices have the full set of technical information available. Different information sources are available for different model lines - while some vendors will publish the firmware update packages, others avoid providing even the changelogs. Certification records completeness depends on the target country where the device is sold. Particular modules implemented in our prototype, such as static application and firmware analysis are based on the predefined keyset and can be extended with more advanced mechanisms based on other semantic constructs and flow analysis to increase the precision of capability identification. To facilitate the interpretation of raw obtained profiles not only for researchers but also for use by private device consumers, the tool can be extended with the simpler, graphical frontend. This would allow for browsing through the device profiles and comparing the potential security-sensitive functionality within IoT products.

VII. Related work

A. IoT characterization: A classification method for IoT devices using a clustering approach [27] is based on relevant information such as size, mobility, memory, bandwidth, WiFi and bluetooth etc. The methodology allows for automatic identification of the brand and model of the device for which the firmware sample is built. In [26], authors presented a solution for firmware characterization using machine learning algorithms. The initial training was based on the online sources. At the next stage, it was possible to classify firmware for a particular device with reasonable accuracy. An interesting application of the semantic profiling approach would be for the interoperability analysis of IoT, as proposed in [22]. The authors present a framework which validates information flow, data consistency, and estimates device interoperability through an analysis of companion mobile applications. This methodology focuses on interoperability in a mutual system, such as event distribution, which affects states of IoT sensors and other implicitly and explicitly connected components. In contrast to this approach, our framework covers a set of multiple information sources to identify security and privacy risks, stimulated by presence of the specific device functionality. Use of extensible criteria is introduced to allow for the large-scale comparative analysis of independent IoT products from different vendors and their ecosystems.

B. Security of IoT: Extensive research is done to reveal in-network IoT device activity by profiling its network activity

[13] https://shodan.io

patterns in [10]. Authors in [18] analyzed major IoT security challenges and problems, and presented a detailed analysis of IoT attack surfaces, threat models, security issues, requirements, forensics and challenges. They also formulated actual open problems in IoT security and privacy. More globally, in [29] it was claimed that the world is heading towards catastrophic consequences resulting from the increasing size of networks with billions of insecure IoT devices. They included discussion on the upcoming challenges and potential precautions that can be implemented. IoTSAT [24], a formal framework for security analysis of the IoT was presented in 2016. A wide attack surface for IoT devices, such as onboard IoT for vehicles [19], raises difficulties of predicting unexplored attacks and vulnerabilities. Thus, IoTSAT, a formal security analysis framework, models IoT devices based on a system of hardware configurations, topologies, user policies, and IoT specific attack surfaces. After building the models, the framework is used to measure resilience against potential attacks. IoTSAT evaluates realistic IoT networks for scalability and applicability. Machine learning - based security solutions for IoT raise additional challenges described in [28], [23] which affect their implementation on practice.

C. IoT privacy: A lot of consumer private data has been reportedly exposed over the years. This includes home activities, workout patterns [14], medical information [21], children data [16] and sexual activities [20]. Federal communication commission (FCC) [13], [11], [4], revised the privacy regulations for such devices in 2017. However, implementation of the given policies by IoT manufacturers is not very active. In addition, FCC policies do not cover other important cases, such as possibility of private data segregation from IoT device network traffic for Internet service providers, IoT device vendors and back-end infrastructure owners [11]. The growing number of cryptography solutions is dedicated to preventing the parties from observing sensitive data in device communication [17].

VIII. Conclusion

This work presented a semantic approach, which enables profiling of existing and upcoming IoT devices to reveal, categorize and compare their security-sensitive capabilities. The proposed methodology facilitates early-stage, low-budget, as well as large-scale analysis of IoT devices without relying on physical access to actual hardware.

We introduced a set of direct and indirect information sources which uncovers technical information about device capabilities, revealing the presence of common built-in hardware and software components. We have developed a software tool prototype and provided two case studies to demonstrate the applicability of the approach in profiling real-world market products. Despite its limitations, the proposed methodology is helpful as a preliminary step for targeted risk analysis and characterization of existing products. It enables profiling in situations where obtaining a physical device is not possible, such as research of upcoming IoT products in the early stage before they find their way to the market.

Finally, the proposed use of extensible criteria sets facilitates targeted large-scale security studies of Enterprise, Medical, Industrial environments and Critical Infrastructures to continuously discover vulnerabilities, which are being introduced to these systems by IoT components

References

[1] Amazon Echo Dot v2 [Online]. Available: https://www.amazon.com/Amazon-Echo-Dot-Portable-Bluetooth-Speaker-with-Alexa-Black/dp/B01DFKC2SO.

[2] FCC ID 2AHSE-2045. [Online]. Available: https://fccid.io/2AHSE-2045/Users-Manual/User-Manual-FCC-3131987.

[3] FCC ID IPH-A2758. [Online]. Available: https://fccid.io/IPH-A2758.

[4] Federal Communications Commission. Office of Engineering and Technology [Online]. Available: https://apps.fcc.gov/oetcf/eas/reports/GenericSearch.cfm.

[5] Forerunner 230 firmware [Online]. Available: http://www.gmaptool.eu/en/content/forerunner.

[6] Garmin Support [Online]. Available: https://www8.garmin.com/support/download_details.jsp?id=9513.

[7] https://medium.com/@micaksica/exploring-the-amazon-echo-dot-part-1-intercepting-firmware-updates-c7e0f9408b59.

[8] Kontakt [Online]. Available: https://kontakt.io/blog/extensive-guide-to-bluetooth-beacons.

[9] OWASP. Firmware Analysis [Online]. Available: https://www.owasp.org/index.php/IoT_Firmware_Analysis .

[10] Eavesdropping attacks on high-frequency RFID tokens, author=Hancke, Gerhard and others. In *4th Workshop on RFID Security (RFIDSec)*, pages 100–113, 2008.

[11] FCC [Online]. Available: https://www.fcc.gov/ document/fcc-adopts-broadband-consumer-privacy-rules, 2016.

[12] IoT goes nuclear: Creating a ZigBee chain reaction, author=Ronen, Eyal and et al. In *IEEE S&P*, pages 195–212, 2017.

[13] F. C. Commission et al. Protecting the privacy of customers of broadband and other telecommunications services, 2016.

[14] T. Davenport and J. Lucker. Running on data: Activity trackers and the Internet of Things. *Deloitte Review*, 16, 2015.

[15] E. Fernandes, J. Jung, and A. Prakash. Security Analysis of Emerging Smart Home Applications. In *IEEE S&P*, pages 636–654, 2016.

[16] L. Franceschi-Bicchierai. Internet of things teddy bear leaked 2 million parent and kids message recordings. *Motherboard*, 2017.

[17] P. S. J. Hemmings and A. Kirkland. The Institute for Information Security and Privacy, 2016.

[18] M. M. Hossain, M. Fotouhi, and R. Hasan. Towards an analysis of security issues, challenges, and open problems in the internet of things. In *Services (SERVICES) World Congress on*, pages 21–28. IEEE, 2015.

[19] T. Huang, J. Zhou, and A. Bytes. Atg: An attack traffic generation tool for security testing of in-vehicle can bus. In *Proceedings of the 13th ARES*, pages 32:1–32:6, 2018.

[20] D. Kravets. Sex toys and the internet of things collidewhat could go wrong, 2016.

[21] N. Lars. Connected Medical Devices, Apps: Are They Leading the IoT Revolutionor Vice Versa.

[22] I. Lee and K. Lee. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431–440, 2015.

[23] Y. Meidan and et al. Detection of Unauthorized IoT Devices Using Machine Learning Techniques. *arXiv preprint arXiv:1709.04647*, 2017.

[24] M. Mohsin, Z. Anwar, G. Husari, E. Al-Shaer, and M. A. Rahman. IoTSAT: A formal framework for security analysis of the internet of things (IoT). In *IEEE CNS*, pages 180–188. IEEE, 2016.

[25] S. Notra, M. Siddiqi, H. H. Gharakheili, V. Sivaraman, and R. Boreli. An experimental study of security and privacy risks with emerging household appliances. In *2014 IEEE CNS*, pages 79–84, 2014.

[26] R. Roman and et al. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266–2279, 2013.

[27] J. N. Surez and A. Salcedo. ID3 and k-means Based Methodology for Internet of Things Device Classification. In *ICMEAE*, page 5, 2017.

[28] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu. IoT Security Techniques Based on Machine Learning. *arXiv preprint arXiv:1801.06275*, 2018.

[29] T. Yu and et al. Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things. In *In 14th Workshop on Hot Topics in Networks*, page 5. ACM, 2015.

[30] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE IoT journal*, 1(1):22–32, 2014.