Towards a Security Classification of IoT and Smarthome Systems

Kevin Löhmann[1], Martin Schröer, Karsten Sohr[2]

Abstract: During the last years the Internet of Things (IoT) has gained a significant market relevance, more and more companies have started to rapidly push devices and systems to the market. As shown by a number of scientific papers and documentations of practical attacks, many of the devices currently available are still vulnerable to well known attacks—often with significant consequences for their users. As of today, there is no commonly accepted taxonomy for structured classification of IoT and smarthome systems available. We propose a comprehensive scheme for classification of such systems based on their security-related properties. For this, we selected a variety of systems currently available to consumers featuring different functional scopes and technical properties and set up a catalogue of criteria. We further refined this catalogue based on reports of known attacks and other scientific works to also cover sparsely documented properties. The resulting classification scheme may serve as a foundation for further works like vulnerability- and best-practice catalogues, threat-modeling, security analysis of specific systems and more. We were able to successfully evaluate the scheme by applying it to a number of systems and their corresponding features, allowing us to identify potential vulnerabilities that would have possibly been overlooked without the help of the novel scheme.

1 Introduction

Presently, the growing IoT market rapidly emerges more and more systems, bringing Internet access to coffee makers, home lighting, doorknobs and more. Most manufacturers focus primarily to fastly gain market shares and renownedness. Security aspects are often subordinated or even ommitted in favor of shorter times to market, thus introducing a plethora of—often well known—vulnerabilities. To provide a foundation to tackle this issue, a taxonomy to categorise IoT devices and smarthome systems to gain an structured overview is needed.

To match this need, we present a corresponding classification scheme, which is intended to serve as a starting point towards a standardised classification of IoT and smarthome systems while focusing on security aspects.

To the best of our knowledge, there is no adequate scheme available that would suit the current needs: only sparse, purpose-related specific classifications are defined and used in respective articles.

[1] neusta mobile solutions GmbH, Konsul-Smidt-Str. 24, 28217 Bremen, Germany k.loehmann@neusta.de
[2] University of Bremen, Technologiezentrum Informatik, Bibliothekstr. 3, 28359 Bremen, Germany schroeer@uni-bremen.de, sohr@tzi.de

Classifications performed this way are not comparable and do not allow to put results into a holistic context as needed for a comprehensive treatment of the issue. Our approach is to define a common, yet comprehensive scheme of security-related properties of smarthome systems and components. It can be applied to serve various purposes:

- Definition of abstract architecture elements to serve as a foundation for formal analysis, e.g., as demonstrated by Tabrizi and Pattabiraman [12]
- Collecting present and future work into a holistic, thus comparable context
- A starting point for certification processes and structured security analysis (e.g. in federal institutions)
- Definition of specific threat catalogues for smarthome applications
- Definition of smarthome specific security architectures

The classification scheme is constituted by distinct categories, which cluster security-related properties of IoT and smarthome systems, which we derived from a market research. A unique characteristic of the classification scheme is it being capable of describing actual systems in a way abstract enough to enable comparison of systems sharing similar properties while on the other hand also being specific enough to describe a particular system in appropriate detail.

1.1 Related work

Riahi et. al. [9] analyse the security related layers around IoT systems and their role in the systems. Their approach focuses on providing a holistic perspective for IoT security in general, but omits properties that are specific to smarthome systems. The IoT Security Initiave (IOTSI) has defined an operational classification for components and devices [10]. This classification focuses on device types but neglects the consideration of entire IoT systems. A model for IoT security based on functional requirements is defined by Barbar et. al. [1]. Their model is general and builds a foundation for more detailed analyses. A comprehensive overview about general security considerations in the area of IoT is provided by Zhang et.al. [13]. Their considerations can be applied in further works to identify vulnerabilities and define specific security requirements for the properties listed in this paper. Tabrizi and Pattabiraman [12] present a formal analysis of an abstract model of a smart meter. They classify systems in the specific context of their study, hence it is hard to transfer their findings to similar devices.

A detailed security analysis of Samsung SmartThings is performed by Fernandes et.al. [5], enabling them to discover security related design flaws. Their approach could be put into a broader context to give hints about security flaws in systems sharing similar features.

Sivaraman et.al. [11] demonstrate attacking a smarthome system by use of a manipulated mobile app. They stress the point that perimeter security is not sufficient, but rather an integrated approach covering all components and their interactions is required. The scheme proposed by us supports this perspective by taking into account whole systems, including all contained components.

1.2 Background

The Internet of Things (IoT) is a still growing market in industry and consumer electronics in which smarthome systems play a major role. Masses of IoT and smarthome systems are rapidly pushed to the market—often flawed with vulnerabilities that could easily be exploited as shown, e.g., by Dhajani [4]. Due to the private, often sensitive nature of the data processed and the possible consequences of its abuse, security should be considered to be an important aspect. Vulnerabilities in existing systems have been actively exploited with various consequences, such as:

- Building partially huge botnet structures to prepare other attacks [6]
- Irreversibly breaking devices in order to make them unusable [8]
- Violating privacy and intimacy by unauthorised surveillance [3]
- Lock picking of smart door locks [2]

Various related works pursue different approaches to evaluate and improve security in smarthome applications. In most of those works a specific classification scheme is defined which is directly related and specifically suited to the topic of a particular study. Yet, there is no general scheme available to classify smarthome systems, components and protocols that would allow one to compare studies and findings.

IoT applications underly specific constraints that do not apply to traditional IT systems. This includes, for instance, usage of constrained devices, high numbers of interconnected devices, auto discovery and in many cases the lack of trustworthy instances to ensure authenticity. Beyond this, manufacturers focus on a non-technical target audience, providing convieniece functions such as easy setup and ease of use, thus accepting trade-offs in security in favor of usability. Therefore, it can be considered to be a viable approach to define a classification scheme tailored towards IoT and smarthome systems, which takes those special constraints into account. This scheme may be generalized to also cover future IoT applications, e.g. industrial controllers, automotive applications and medical devices.

2 Classification Scheme

The proposed classification scheme is constituted by six categories describing security related aspects, which cluster particular properties and characteristics of IoT and smarthome systems that are significant for security considerations:

- Component Types
- Automation
- Extensibility
- Access Paths
- Technical Characteristics
- Functional Scope

In the following sections we will describe the particular properties we selected for classification of IoT and smarthome systems. In doing so, we will also provide an overview of common components and technologies used.

2.1 Component Types

Components are the physical parts that constitute a system, e.g., actuators, sensors and controllers. In our research we found several types of recurring components. From the security perspective, the component types present in a particular system provide an indication of possible attack vectors. Independent of the attack vectors for specific components, generic attack vectors may apply to every component type. Furthermore, the more component types are used in a system, the more attack vectors are probably to be found in the interaction layer amongst them.

There may also be compound devices in a system, to which several component types apply simultaneously: in this case, all component types that apply to those devices must be selected for a appropriate classification of the respective system.

Component type	Description
Sensors	Sensors are passive components that signal environmental conditions to the system but do not manipulate their environment. Sensors may receive commands, e.g., for device configuration. Forging sensor data may enable to provoke unauthorised reactions of the system.
Actuators	Actuators are active components which manipulate their environment based on commands from other components. Providing actuators with forged commands may enable unauthorised control.

Component type	Description
Cameras	Cameras are more complex than other sensors as they process high amounts of data and usually require special communication behaviours and protocols. Compromising cameras may provide unauthorised surveillance, which may lead to various possibilities of abuse.
Repeaters	Repeaters are passive components which forward messages between components using the same communication protocol. Faking identity of or infiltrating a vulnerable repeater may enable one to eavesdrop on an arbitrary amount of a system's communication or to tamper with it.
Protocol Bridge	Unlike repeaters, protocol bridges mediate communication between different protocols, in contrast to controllers, they do not translate the information payload but encapsulate it.
Controller	A controller is a component which uses complex logic, e.g., to manage components in one or multiple component networks, execute automation rules, provide API for frontend applications, connect to external services and more. Hence a controller is considered to be a complex device, which may expose a multitude of vulnerabilities.
Mobile App	Mobile apps are usually used to give a user convenient access to a system's functionality. This may include viewing and modifying components' status and system configuration. As mobile apps are specific to a device executing them, e.g., a tablet or a smartphone, they may be compromised by attacks that the particular device is vulnerable to.
Web frontend	Similar to mobile apps, web frontends give a user access to a system's functionality. In contrast to mobile apps, web frontends are not specific to a particular device and can usually be accessed by any device that is connected to the corresponding network, thus potentially exposing respective vulnerabilities.
Internet Backend	Internet backends provide server- or cloud-based Internet access and may enable one to outsource and backup a system's data and configuration. Forging an Internet backend's identity or tampering with the communication between the backend and a system may provide arbitrary access to the connected system.

Tab. 1: Component types of IoT and smarthome systems

2.2 Automation

Many systems provide a functionality to automate processes described by rules—for instance, actuators performing actions based on corresponding sensor inputs. In general, a higher degree of automation may raise more potential vulnerabilities.

Automation Type	Description
Sensor-Actuator Association	In sensor-actuator association, a sensor's output is directly associated to an actuator's input, i.e., the actuator will perform an action if the sensor emits a corresponding value. This requires associated actors and sensors to use the same protocol and may remain active even in absence of a controller or communication with frontends. The often simple point-to-point communication between associated devices may be tampered with.
Remote Control	In the scheme, remote control is defined to be performed through web frontends or mobile apps. In contrast to sensor-actuator association, components to be controlled usually use various different communication protocols, requiring a controller or bridge to communicate with.
Static Rules	Static rules are usually trigger-based, using, e.g., sensor values, user interaction or time conditions to execute corresponding actions. While the evaluation of static rules is usually performed by a component with higher logic functions, e.g., by a controller or Internet backend, and may include simple calculations to generate output values, static rules do not include variable arguments, which would turn them into adaptive automation.
Adaptive Automation	Adaptive automation is based on static rules with variable parameters that are evaluated and refined by a learning function, which can potentially be tampered with.

Tab. 2: Possible automation types

2.3 Extensibility

IoT and smarthome systems may be extended in various ways. From the security perspective extensibility requires either functionality that is not used if the extension is not available or relies on interfaces (software or hardware), which may expose vulnerabilities. Extensibility also refers to components that can be used within the system. A fully extensible system would allow the addition of arbitrary devices, while a closed system would only accept integration of a number of specific trustworthy devices.

Extensibility	**Description**
Extension with Arbitrary Devices	While most IoT and smarthome systems favor specific extension devices, e.g., only supporting devices of the system's manufacturer, some systems also allow for the extension with arbitrary devices, providing such devices with at least a minium set of functionality. Application of arbitrary devices may cause the introduction of vulnerabilities specific to a particular device.
Third-party Services	External services usually use REST APIs to integrate their functionality into a system. A common way is to use webhooks or foreign APIs, while a system itself may also provide a dedicated API for third-party use. In the scheme definition this also includes non-public APIs. As network APIs can be reverse engineered, it is irreleavant whether a documentation is publicly available or not.
Integration with other Smarthome Systems	Many systems allow the integration of other IoT and smarthome systems, for instance, it is possible to include Phillips Hue with many other systems. If this integration is performed, the host-system will inherit at least some of the vulnerabilities of the connected system.
Software Add-Ons and Firmware Upgrades	Systems that allow one to install add-ons (plugins) or to update the firmware are regarded equally vulnerable from the security perspective, as both enable one to install potentially harmful software.
Hidden Features	Some systems offer additional features that may be unlocked with a special token, e.g., an unlock code. In contrast to extensibility by software add-ons, such hidden features are usually already present in the system, potentially inlcuding additional, not obvious vulnerabilities.
Hardware Add-Ons	Hardware add-ons enable one to provide components with additional functionality, e.g., dongles to support additional protocols. Unlike arbitrary devices which depend on existing functionality, hardware add-ons provide extra functionality, potentially exposing additional attack vectors to the system. To enable the integration of such add-ons, the component to be extended usually already contains corresponding software.
Extension Modules	Extension modules accumulate the characteristics and security considerations of both soft- and hardware add-ons, i.e., hardware along with the software needed for its operation within the target system. This may also entail further vulnerabilities.

Tab. 3: Extensibility Types

2.4 Access Paths

Access paths describe the ways the user can interact with the system and how the components interact amongst each other.

Access Path	Description
Radio components	Many systems use radio protocols like IEEE 802.15.4, Bluetooth or WiFi for communication between components. This may allow an attacker to manipulate the system from some distance without having to be present at the location.
Wired Components	In wired components, communication between the components is performed via cable instead of RF. For an attacker, physical access to the cable is required to perform malicious actions. Based on this fact, some wire-based protocols entirely lack authentication and encryption.
Local network API	Frontends often may use an API over the local network to communicate with components. This access path may be missing network security features that are not usable in local environments (like TLS). Furthermore, many systems seem to rely on local networks as a trusted connection.
Internet API	The system exposes a network API for frontends or external services which makes it accessible via Internet. In this scenario the possible group of attackers is larger, but it is also possible to incorporate more security features compared to a local environment.
Non-API access	Commonly, systems provide an API to access the specific functionality of the system, some systems also provide non-API access, e.g. to upload or download files via ftp.

Tab. 4: Access Paths

2.5 Technical Characteristics

Besides general security considerations that may arise from properties in this catalogue, technologies used in a system may also expose particular vulnerabilities. Therefore specific security considerations have to be taken into account. To address this in an assessment and further verification, all technologies used must be collected. Based on such a collection, a specific catalogue of threats and measures for the used technologies can be established for a system to be evaluated. If not all details about technical characteristics are available in the assessment, it is viable to assume the highest applicable threat level.

2.5.1 Component Software Stack

This section lists all software parts used in the components, including firmwares, SDKs, operating system components (e.g. kernel and libraries), application platform (e.g. Java, OpenHAB) and application libraries. Every software part is subject to specific security considerations and may raise specific vulnerabilities, e.g., heartbleed [7]. Vulnerabilities may also arise from wrong usage of the libraries and SDKs in use. Examples of component software stacks are:

Category	**Examples**
Controller Software	Linux kernel version x.y.z, Linux Operating System, Libraries used in controllers software stack, Application software, Software platform for application platform (e.g., java 1.8), CoAP Library
Constrained Devices	Chip firmware (e.g. Atmel firmware version), Z-Wave SDK incl. version, MQTT library and version, CoAP library incl. version
Android App	Android SDK-Level, Butterknife, Dagger 2
iOS App	iOS SDK version, Typhoon

Tab. 5: Examples of items in a software stack

2.5.2 Communication Protocols

In this section all protocols and protocol characteristics (like version and encryption) used for communication between components or foreign services must be listed by the corresponding communication partners. Hence this list must contain all devices in the system and also cover all possible communication relations between components. Examples of communication protocols are:

Protocol	**Communication Path**		
HTTP 1.1	Controller	$\longleftrightarrow$	Hue Bridge
	Frontend	$\longrightarrow$	Backend
	Backend	$\longrightarrow$	IFTT
TLS 1.2	Frontend	$\longrightarrow$	Backend
	Backend	$\longrightarrow$	IFTT
OAuth 2.0	Frontend	$\longrightarrow$	Backend
ZigBee Light Link	Hue Bridge	$\longleftrightarrow$	Hue Lamp

Tab. 6: Communication Protocols

2.6 Functional Scope

In this section all supported functional domains of the system are to be listed. Some systems have a single purpose, others provide a variety of functionalities at the same time. In general, a system supporting many functional domains may expose more vulnerabilities than a system with a narrow functional scope. Also some functional domains may expose higher security risks than others, thus the functional domains can be mapped to threat classes. Examples for functional domains may be: Lighting, Motion detection, Door lock, Irrigation or Smoke detection.

3 Conclusion

We propose a unified scheme for classification of IoT and smarthome systems. Based on an analysis of the security characteristics of 23 selected smarthome systems, we were able to sucessfully evaluate the proposed scheme for its general applicability. Furthermore, the exemplary application of the proposed scheme to a number of selected systems (Bosch Smart Home, IKEA Tradfri, Telekom Magenta SmartHome) enabled us to identify similarities between the compared systems indicating shared vulnerabilities, which probably would have been unnoticed if the concerned systems had been evaluated separately. The classification scheme provides a basic structure for an analysis of additional specific systems and gives first ideas of vulnerabilities possibly present in a system. It also allows for the integration of further system-specific reports into a common knowledge base, significantly facilitating the application gathered from a specific system to others sharing similar features.

The classification scheme will not substitute for a detailed and in-depth analysis of systems and searching for unknown vulnerabilities in specific systems, but it will facilitate transfering knowledge about known and novel vulnerabilities between systems with similar properties. As the current classification scheme is focused on smarthome systems and IoT components, it may need further refinement to be also suitable for other IoT applications such as industrial or automotive environments. As the catalogue presented in this paper is derived from market research based on a selection of 23 systems—it may not cover all aspects of systems available, therefore it is subject to review and justification based on further research.

Currently, we are preparing an online tool based on the classification scheme that will enable visitors to conveniently perform classification of arbitrary IoT and smarthome systems, resulting in a report highlighting the security-related characteristics of the specified system, while also provding specific recommendations and best practices for improving security. Beyond this, the information gathered will be used for expansion and refinement of a knowledge base about IoT and smarthome system characteristics and their distribution.

References

[1] Sachin Babar et al. “Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT)”. In: *Recent Trends in Network Security and Applications* 89.Chapter 42 (2010), pp. 420–429.

[2] Nitesh Dhanjani. “2. Electronic Lock Picking—Abusing Door Locks to Compromise Physical Security - Abusing the Internet of Things”. In: *Abusing the Internet of Things*. O’Reilly, Aug. 2015.

[3] Nitesh Dhanjani. “3. Assaulting the Radio Nurse—Breaching Baby Monitors and One Other Thing - Abusing the Internet of Things”. In: *Abusing the Internet of Things*. Aug. 2015.

[4] Nitesh Dhanjani. *Abusing the Internet of Things*. Blackouts, Freakouts, and Stakeouts. O’Reilly Media, Aug. 2015.

[5] E Fernandes, J Jung, and A Prakash. “Security Analysis of Emerging Smart Home Applications”. In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2016, pp. 636–654.

[6] Dan Goodin. *Record-breaking DDoS reportedly delivered by >145k hacked cameras*. [Online; accessed 15-July-2017]. Sept. 2016. URL: https://arstechnica.com/security/2016/09/botnet-of-145k-cameras-reportedly-deliver-internets-biggest-ddos-ever/.

[7] Synopsys Inc. *Heartbleed Bug*. [Online; accessed 15-July-2017]. Apr. 2014. URL: http://heartbleed.com/.

[8] Douglas Jose Pereira dos Santos. *Bricker Bot – A Silver Lining to Force Accountability for IoT Security?* [Online; accessed 15-July-2017]. May 2017. URL: http://blog.fortinet.com/2017/05/02/bricker-bot-a-silver-lining-to-force-accountability-for-iot-security.

[9] A Riahi et al. “A systemic and cognitive approach for IoT security”. In: *2014 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, Feb. 2014, pp. 183–188.

[10] MI-SF LLC. *IOTSI - Framework*. [Online; accessed 15-July-2017]. URL: https://www.iotsi.org/.

[11] Vijay Sivaraman et al. *Smart-Phones Attacking Smart-Homes*. New York, New York, USA: ACM, July 2016.

[12] Farid Molazem Tabrizi and Karthik Pattabiraman. “Formal Security Analysis of Smart Embedded Systems”. In: *Proceedings of the 32Nd Annual Conference on Computer Security Applications*. New York, NY, USA: ACM, 2016, pp. 1–15.

[13] Zhi-Kai Zhang et al. “IoT Security: Ongoing Challenges and Research Opportunities”. In: *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications (SOCA)*. IEEE, 2014, pp. 230–234.