International Journal of Information Security (2020) 19:3–14
https://doi.org/10.1007/s10207-019-00445-y

SPECIAL ISSUE PAPER

The IoT security gap: a look down into the valley between threat models and their implementation

Peter Aufner[1]

Published online: 19 June 2019

Abstract

We claim to have identified gaps between threat modeling frameworks, threat model use in IoT security research and attacks that may be missed by current research. While security research includes sections known as 'threat models', these models are not supported by the categorization and standardization that threat modeling frameworks would have to offer. Then again, if existing threat modeling frameworks were used, they would still allow many vulnerabilities to pass through undetected, since they are meant for software-only projects. This work will explain the origins of IoT research, enumerate common threat modeling frameworks and give an insight into the state of IoT security research. In the course of this, it will become clear how these gaps came to be and what research directions would help to close them.

Keywords Internet of Things · Threat modeling · STRIDE · CORAS · Privacy · Security · Vulnerabilities

Abbreviations

DoS	Denial of Service
SDL	Security Development Lifecycle
WSN	Wireless Sensor Network

1 Introduction

Consumer IoT is a hot IT security research topic right now. Attacks against devices and methods are being devised; defenses against these potential attacks are being developed and brought into place while overall device security is under constant scrutiny.

Despite so much research being carried out, we have identified three gaps left to fill. This work will shed light on these gaps. To do so, we will first introduce threat modeling frameworks as a base for systematic IT security research. We will then give a brief overview of definitions by which IoT devices are categorized from a research perspective. This will be followed by a focus on the frequently mentioned sample attacks during IoT security evaluation and by some examples of how threat modeling is carried out in the context of IoT papers.

Based on these introductions and definitions, we will explain the gaps we have identified between what threat modeling frameworks offer, how IoT security researchers use threat modeling and what IoT security research leaves out. These can be seen both as hints for new research topics and also as a perspective extension for IoT security.

2 An overview of threat modeling frameworks

This section gives an introduction to four different threat modeling frameworks.

Threat modeling is an established way of enumerating and categorizing potential security implications for a given technology. In the field of computer science, there are several mature frameworks that offer guidance through the process. These frameworks were designed with software in mind, and their applicability to IoT may thus be limited. We examined sources relevant for possible threat modeling frameworks in the context of IoT. We have ultimately based our choice on the selection made by [1] who have enumerated a broad range of frameworks, among which some of the most cited ones appeared. The following paragraphs shall provide a brief glimpse of the versatility these frameworks display.

STRIDE is by far the most frequently cited framework. We found it being used for e-banking applications [2] down

✉ Peter Aufner
peter.aufner@iaik.tugraz.at

[1] IAIK - Institute for Applied Information Processing and Communications, Inffeldgasse 16a, 8010 Graz, Austria

Table 1 A list of threat modeling frameworks as enumerated in [1]

Framework name	Base artifact	Automated tool	Based on
STRIDE	Data-flow diagram	No	Original
LINDDUN	Data-flow diagram	No	Original
CORAS	UML diagrams	Diagram editor	Original
Quantitative threat modeling	Data-flow diagram	No	
Abuser stories	Data-flow diagram	No	STRIDE
STRIDE average model	Data-flow diagram	No	STRIDE
Attack trees	Attack trees	SecureI tree	Original
Fuzzy logic	Data-flow diagram	MATLAB fuzzy logic toolset	STRIDE
T-MAP	UML diagrams	Tiramisu	Original

to the level of Software Defined Networking [3]. The authors of [4] used STRIDE in the context of Internet of Everything (IoE).

Quantitative threat modeling was mainly picked as a useful extension to STRIDE and also to show that there is innovation in the field of threat modeling frameworks.

CORAS is also often cited in academia. This framework appears, for example, in the context of network embedded systems [5]. It is also applied to study communication security between smartphones and IoT devices [6]. The approach is of interest because it is meant for consulting, opposed to the other approaches that are meant to be used by developers.

LINDDUN was chosen for its focus on privacy. While the previous three frameworks only deal with security threats, this framework focuses on privacy as opposed to the security focus of the other frameworks. The framework was named in a paper that attempted to create a privacy analysis framework for IoT as not properly applicable [7]. However, LINDDUN has been applied in the context of smart grids [8]. It was also mentioned in the context of privacy by design [9] and as a support for achieving GDPR compliance [10].

We have picked the threat modeling frameworks based on diversity in purpose, the goals they aim to fulfill and completeness in terms of how much of the whole process they cover, from identification to mitigation.

For a quick overview of common threat modeling frameworks, the base artifacts they build on, whether there is tool support available and if they build on another framework, see Table 1. The following subsections explain how to apply the most popular frameworks in brief.

2.1 STRIDE

STRIDE was developed by Microsoft and integrates with its Security Development Lifecycle (SDL), and it is thus one of the most popular threat modeling frameworks for software engineering [11]. STRIDE uses data-flow diagrams that are produced as part of the SDL. These diagrams are used as mappings to identify threats.

STRIDE is an acronym that stands for:

- Spoofing
- Tampering
- Repudiation
- Information Disclosure
- Denial of Service
- Elevation of Privilege

The data-flow diagrams should already be present as artifacts of the software engineering process. They are designed to reveal the interactions between complex processes, (simple) processes, external entities and data stores all of which are connected via data flows. To make them useful for threat modeling, they have to be extended with privilege boundaries that indicate whether data is crossing privilege boundaries. The complex processes are processes that consist of multiple elements. They must further be dismantled until only the other primitives of the framework are left.

After all simple primitives are revealed, they are assigned numbers. Data flows are enumerated by the numbers of the other elements they connect.

The framework provides a mapping of element type to potential threat categories. For example, a data flow may be susceptible to tampering, information disclosure and Denial of Service. There is also a mapping of threat categories to risk levels provided. For example, targeted information disclosure on the server side by reading from known locations is assigned a risk level of three. In this case, higher levels mean greater severity. Alternatively, the DREAD model is also commonly used to assign ratings to threats.

Based on these assignments, it is up to the users of the framework to decide what sort of mitigation, if any, they decide to apply.

The process of using STRIDE is completely manual [12].

2.2 LINDDUN

LINDDUN is defined as a framework to model privacy threats in software-based systems. Like STRIDE, which the framework is modeled after, LINDDUN uses data-flow diagrams.

It categorizes privacy threats in the following categories:

- Linkability
- Identifiability
- Non-repudiation
- Detectability
- Disclosure of information
- Unawareness
- Non-compliance

Similar to STRIDE, LINDDUN also provides a mapping table of potential threats to the various kinds of elements in the data-flow diagram. After the links are established, threat tree patterns are provided to further specify the implication of a threat to the implementation of the element it is assigned to. The information gathered up to this point in the process is refined into misuse cases. These are the same as use case scenarios, but now seen from an attacker's point of view.

Any risk assessment technique can be used for assigning risk values to the identified threats. This could also be DREAD as it is used in STRIDE or OWASP's risk rating methodology.

Finally, LINDDUN does not provide suggestions on how exactly to mitigate risks but suggests generally the use of technical or legal means to enhance privacy [13,14].

2.3 CORAS

CORAS works in an iterative process between analysts and developers during software development. There are seven steps that focus around meetings between clients and analysts. In this regard, it is different from the other frameworks introduced so far, since CORAS centers around the idea of analysts and developers being distinctively different persons. CORAS is done in seven steps:

1. The goals of the analysis are set, and information on the target is gathered.
2. The analysts present their understanding of the target and also a high-level security analysis. Also, at this point the threats, vulnerabilities, threat scenarios and unwanted incidents are identified.
3. Additionally, at this point all the information gathered so far is presented and approved by the client.
4. A workshop is held with people from various fields of expertise. This should help in identifying as many unwanted incidents as possible.
5. An additional follow-up workshop is held to estimate the consequences and assign likelihood values to the unwanted incidents.
6. Subsequently, an overall risk picture is presented to the client and adjusted.
7. Mitigations are identified as well as a cost/benefit analysis on how to resolve the threats.

UML diagrams are used as a basis to illustrate the system under scrutiny, and in subsequent steps are evolved into CORAS diagrams. In this approach, there are several types of diagram in use: the UML class diagram, the UML collaboration diagram and the UML activity diagram. CORAS is more flexible in how it is applied to various projects since it does not automatically imply the allocation of specific threats to specific application types or to elements of them. The focus is rather on the individual treatment of threats based on the project at hand [15].

2.4 Quantitative threat modeling

The authors of the Quantitative Threat Modeling approach have innovated on STRIDE as introduced in Sect. 2.1 by quantifying security and privacy risks associated with the elements of the generated attack trees. This helps to prioritize the mitigation process. Their approach works under the assumption that DREAD was used to qualify threats. They then take into account the concept of damage potential and affected users to add the privacy dimension to the analysis. Discoverability, exploitability and reproducibility are represented as conditional probabilities that add up to an overall probability rating. The security and privacy risks are aggregated inside the attack trees from the leaf nodes up to the root nodes. After completion of this process, it is possible for each previously identified attack to reason about the impact and likelihood. Once the initial attack trees have been built an iterative process of choosing mitigation techniques begins until each risk is either avoided, optimized or completely accepted. This is an advantage since it builds the base for automating the process [16].

2.5 Other approaches

The survey [1] lists a series of other techniques enumerated in Table 1. These will not be discussed in this work. Basically, they are either extensions to one of the methods mentioned above, like the Fuzzy Logic approach [17] which focuses on input fuzzying. On the other hand, attack trees, as introduced by Bruce Schneier, are a very basic approach that is of very little help in categorization and even less help in finding ways to mitigate the threats. Abuser stories [18] are strongly focused on the agile approach and are thus only applicable if the project environment is suitable.

2.6 Summary

In this section, we have introduced various popular threat modeling frameworks. They are all well suited to software engineering projects to the extent that they can seamlessly be integrated into production. These approaches rely on artifacts common to software engineering projects like UML models or interaction between consultants and software developers. The frameworks also have different focus types depending on the type of security enhancement they aim to provide. Some focus mainly on the direct security impact while others put privacy concerns to the fore.

Their tight integration with software engineering is also the base for one gap since this implies a lack of applicability to hardware-centered projects.

In the next section, we will introduce three models for IoT devices each originating at different times and created with different motivations. All of these establish that the requirement for a thing to be categorized as IoT is that there must always be a physical thing combined with internet technology.

3 Definition of IoT devices

In this section, we will discuss three definitions of what makes an IoT device. The term IoT was first used in the context of supply chain management in 1999 [20]. Depending on the point of view and the level of detail at which one examines these devices, they identify different layer quantities that are required to fit the definition. The models we have picked are the following:

1. Three-level model
2. Five-level model
3. CISCO's seven-level model

The models were devised at different stages in the maturity of IoT and come from academia and industry. They thus provide an interesting insight into how the definitions have become more sophisticated over time and how they are based on the point of view of the creator.

3.1 Three-level model

The three-level model is among the first formal models to be defined in [21]. The view on IoT is as an extension to Wireless Sensor Network (WSN). The extension is that the cloud is also provided as an integral part of the model. This model is very simple and general. It already contains the key components of the IoT of today, but it would still fit devices that are not necessarily considered IoT, such as surveillance cameras with cloud integration. There is no mention of collaboration between the devices or their clouds to allow for more complex applications that could not be handled by one specific device alone. For the home sector, the vision has already been called up of centralizing control into, i.e., a smartphone for controlling home appliances or to enable better healthcare.

3.2 Five-level model

The five-level model was introduced in [22]. It is based on the value creation layers of abstract IoT applications observed in academia and in practice.

The concept of the ‘thing’ is more established in this model, since the base layer is identified as a physical thing, such as a light bulb, which was already in existence and is now to be extended. At the second layer, an abstraction is now brought in, and based on the sensors the device comes equipped with, local services such in the case of the light bulb the capability for sensing the presence of people in the room. The third layer is about connectivity in the sense that manageable services are created. This does not include the existence of simple apps to control the things but rather the possibility to send control packets on a network level. The next layer adds analytics to the gathered data thus enabling service composition. The fifth layer represents what the user finally sees at the end of the process. An application, like a smartphone app, is used to control the IoT device.

3.3 CISCO's seven-level model

CISCO has established its own reference model for IoT, which has emerged direct from the industry [23]. The intention is to make a clear distinction between regular networks and IoT.

The first level describes the regular devices. These can be simple sensors but also more complex devices like cameras. The second level is the connectivity both between devices and across network boundaries. Level three is the first computational step. While still in the home network, tasks like basic evaluation or reformatting of data are performed. Level four is about data accumulation. At this point, the data from various devices or networks are processed and stored. Furthermore, a change in paradigm can also be observed at this point as the computing model is converted from event based to query based. At level five, the large amounts of data from the previous levels are abstracted and filtered for easier analysis. Level six provides applications to control or analyze the data gathered so far. Level seven introduces collaboration between users and processes by the IoT devices. This final level can be composed of various sources and adds complex interactions (Fig. 1).

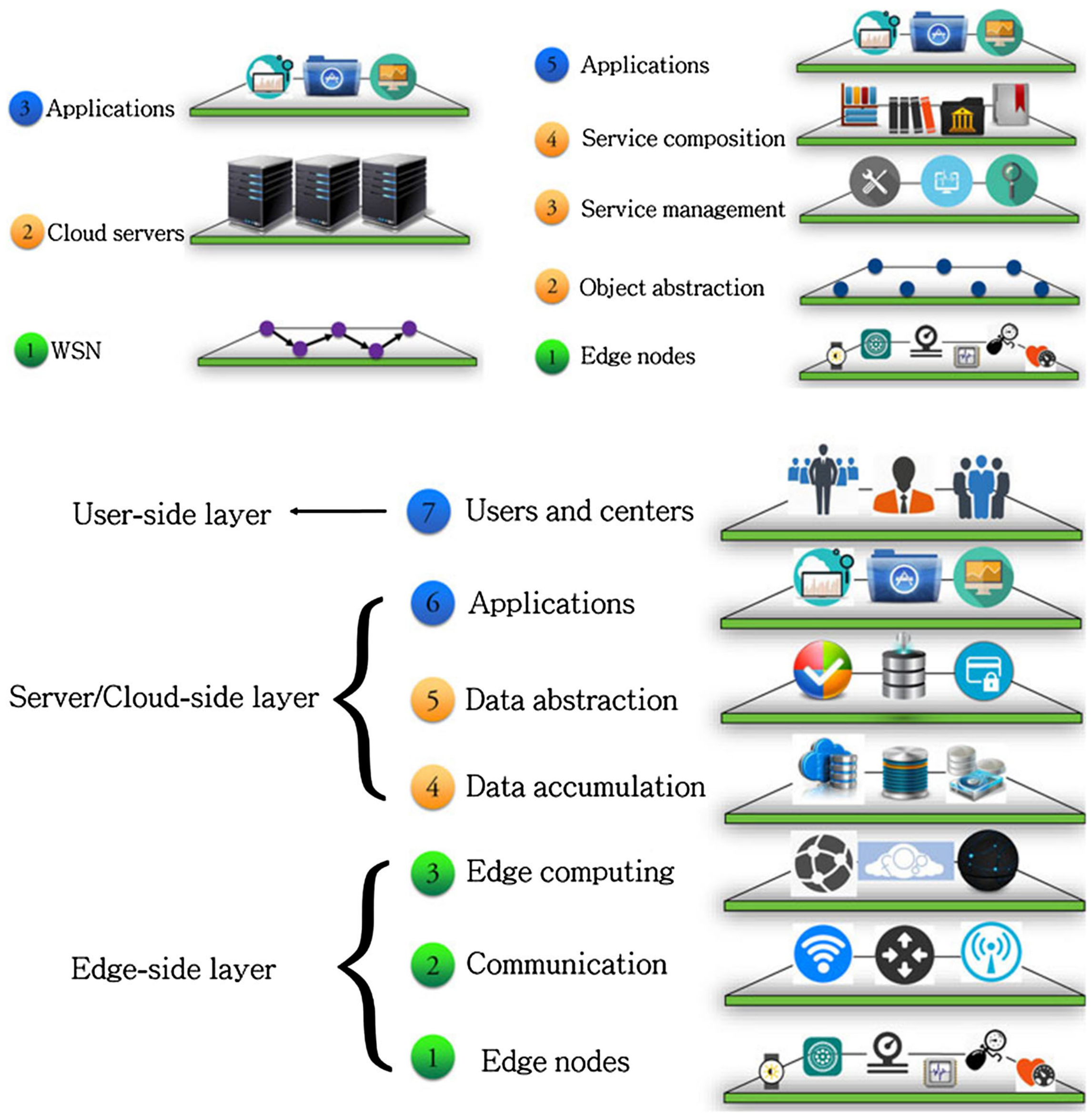

Fig. 1 Three models for IoT architectures [19]. In the upper left, there is the three-level model, upper right is the five-level model and at the bottom is the CISCO seven-level model [19]

3.4 Summary

This section introduced three different definitions of what makes an IoT device. They were derived at different times, with the three- level model being the oldest one, and with different motivations. The five-level model comes from academia and focuses on value creation in each level. CISCO's model has emerged from the industry and is primarily interested in the data generation and processing involved in the making of an IoT device. The important lesson shared by them all is the fundamental idea of having some physical device as the base, that is connected to remote servers which enable the use of applications.

Despite all the definitions recalling how all the IoT devices are based on some physical device, they nevertheless care very little about the implications of a given device. This leads to research focusing on all the other layers while ignoring the basic functionality of the devices they are running on. The next section will provide insights into the current state of IoT security research. Here, we will discuss some common attacks and give an overview of current research directions.

4 Security research in IoT

This section discusses publications in IoT security research. We will examine works both in offensive and defensive researches as well as taxonomic research.

As mentioned in the discussion on various definitions in Sect. 3.1, the origin of academic research in IoT is grounded in WSN. Attacks that were already identified in the context of WSN, especially on the network layer, are applied to different IoT devices. The following four attacks were already examined in the context of WSN [24]:

1. Denial of Service (DoS)
2. Hello Flood
3. Sybil
4. Sinkhole

It will therefore come as no surprise that network-based attacks, especially DoS and the three other attacks above, are cited in surveys as main concerns for IoT [25,26]. We have chosen these four attacks as examples since they are grounded in the most basic layer of IoT, the communication layer, and are thus applicable to any kind of IoT device with a network connection. They have been mentioned in various surveys during almost a decade of WSN and IoT security research. One survey from 2009 [24] already lists them in the context of WSN. They also appear as common and basic attacks in surveys on IoT security published in the year 2016 [19,27,28]. These attacks still appear in surveys published in 2018 [29]. Thus, it stands to reason that they have long been relevant and are still relevant today at the most basic IoT layer.

The surveys mention many, more sophisticated attacks that rely on various levels of device complexity but for the sake of keeping this work as general as possible, we will refrain from using them as examples.

4.1 Four common attacks against IoT

This section explains the previously introduced four attacks in greater detail.

Denial of Service (DoS) has been defined as an event that diminishes or attempts to reduce a network's capacity to perform its expected function [30]. These attacks can either abuse specific functions of another piece of software or hardware or can work by overloading the network with meaningless traffic.

Hello Flood exploits the assumption that when a 'Hello' packet is received the sender must be in close range of the receiver. By using a high-powered transmitter, this can be used by an attacker to make a large number of nodes believe they are in short range of her [31].

The Sybil attack was originally intended to defeat redundancy mechanisms in distributed data storage systems or peer-to-peer networks [32] and was later adapted to WSN [33]. The attack utilizes multiple identities anywhere the concept of 'voting' is used. These identities are used to overturn the algorithm in favor of an outcome desired by the attacker. In the case of WSN, this means that traffic is routed to a malicious node.

The Sinkhole attack works by forging routing information to make a compromised node look more attractive than others. More traffic is thus routed through that node allowing the compromised node control of the data flows in the WSN [31,33].

The common feature of all these attacks is that they work on the network level. Hello flooding can be regarded as a form of DoS. Sybil and Sinkhole are meant to gather traffic without a specific purpose at this stage. They can both be used for DoS but may also be extended into more complex attacks if other weaknesses are exploited.

4.2 An overview of offensive and defensive IoT security research

This section lists publications that discuss attacks on IoT in both an offensive and defensive manners. They are intended to give a feel of what kind of security research is being performed in IoT and to provide a broad overview of the topic.

4.2.1 Defensive research

The authors of [34] introduce 'Pulse', an intrusion detection system they have created. They employ machine learning technologies and claim to identify network scanning and simple forms of DoS. Their results show that the system is actually capable of identifying network scanning activities as well as flooding attacks to a certain degree.

In [35], the application of Recursive InterNetwork Architecture, or RINA in short, is discussed. As the name suggests, recursion is used to stack various policies, chain them or use them side-by-side. This allows for segmentation of the devices and to heavily regulate how they communicate with each other.

Again in defensive research, the authors of [36] have applied the fog computing paradigm to IoT. Their goal is to detect attacks, especially rouge nodes on the network and generally increase trust and privacy in the network.

The authors of [37] postulate that established, host-based security measures, like anti-virus software, will no longer be applicable in IoT. They therefore propose a system based on abstraction that is able to learn attack profiles and normal profiles and work in a dynamic setting to enable context-aware enforcement of policies.

The authors of [38] propose to switch the default networking mode for IoT devices from on to off. They introduce a rule-based framework that enables commands in natural language for defining network conditions at various points in time. One example is to allow the lights to see the luminosity of the bedroom sensor at any given time.

In [39], three attacks against smart locks are identified. One is against state consistency which relies on the fact that smart locks use the smartphone unlocking them to communicate with the internet for authorization. The second one is unwanted unlocking which relies on a device with correct access permissions entering range. In this case, the lock will open automatically. The third attack is privacy leakage. Their solution is to use eventual consistency for access decisions and secure defaults in case of unavailability of servers. To prevent unwanted unlocking, they suggest techniques such as geo-fencing and touch-to-unlock.

4.2.2 Taxonomy and enumeration

In [40], the focus is again on potential attacks against the network side of IoT devices. The authors have performed extensive research on potential attacks at all levels from hardware to software and concluded that blockchain technology can be employed to authenticate devices, secure communication and manage access.

Hossain et al. devised an attack taxonomy for IoT in [41]. The taxonomy is based on a security landscape in three dimensions: connectivity, device specification and area of application. The underlying assumption is that the attacker is tech savvy and thus focuses on either network-based attacks or on manipulation of the device in a high-tech manner, i.e., by modifying the firmware.

The authors of [42] have also focused on network-based threats studying the problem of IoT botnets and have enumerated the main drivers for this rising threat. They have listed several factors among which are the heterogeneity of IoT devices in smart homes, things that were simply not designed to be internet connected and a general lack of ways and means to enforce permissions on the network for the devices. Some basic defense strategies are also proposed such as disabling access to known vulnerable ports and applying updates to the device firmware.

The software engineering perspective was examined in [43]. The authors looked at three broad groups of challenges: methodological challenges, organizational challenges and technical challenges. The development process on how to deal with the legacy code is in focus in the context of methodology. They also establish the pattern of the first thinking about security controls. In the organizational category about the issues are human factors and regulation. From a technical point of view, the focus is on the network-related aspects and the complexities in data flows.

In [44], the three-level model as introduced in Sect. 3.1 is taken as a basis for analysis of security concerns. The authors created an extensive table listing known problems on each of the levels and possible solutions. The view on the IoT devices as networked devices is given once again.

4.2.3 Offensive research

The concept of passive eavesdropping on network traffic in an IoT based smart home is described in the publication ‘A Smart Home is No Castle’ [45]. The authors revealed that despite encryption, simple timing is often sufficient to deduce precisely what and when the inhabitants of a smart home are doing.

Wood et al. examined a clear text data transmission vulnerability in medical IoT devices [46]. In their work, they studied four devices, one of which even leaked sensitive health information. Their second contribution is a user-friendly way to include traffic capturing into a home network.

Also, in offensive research, the authors of [47] have examined the SmartCfg provisioning protocol and found that several vendor implementations are vulnerable to Wi-Fi password theft.

The authors of [48] have examined the security of August smart locks with a focus on the smartphone app. Because the way secrets are stored, it is possible to leak secrets on a rooted device which allows attacks like personal information leakage or DoS.

4.3 Summary

In this section, we have given an overview of the security research being performed in the field of IoT. In the issue of general research, the influence stemming from the original view on IoT as WSN devices is still very strong. The focus is on the network communication aspect and to some degree on the limitations brought about by the small computation power as well as the limited energy supply. Yet, even when single products, like smart locks, are examined, they are treated primarily as networked computing devices with other relevant attacks missing, such as what could physically be done to the lock and how would that impede functionality.

Taking the conclusions from the previous chapters into consideration, it is only fitting that IoT security research should center around network-based vulnerabilities, while basic attacks that affect the underlying hardware are left out.

After establishing the concept of threat modeling and the directions in IoT security research, we will next examine how well threat modeling is integrated into IoT security research.

5 Threat modeling in IoT security research

In the previous section, we introduced a variety of both offensive and defensive research schemes as well as efforts to create taxonomies and to enumerate threats. When examining the security of a device or a category of devices, it is common practice to also provide the underlying threat model. This provides a context for the reader and also helps to establish the relevance of the work. The threat model is commonly described in a section of the paper and provides a context about who the authors have in mind as potential attackers or against whom they intend to provide defense. As introduced in Sect. 2, mature frameworks exist to describe and categorize security and privacy threats, at least on the software level.

We have picked three papers from the works introduced in Sect. 4.2 that illustrate different approaches to security research and examined the provided threat models. The three papers were chosen for their representation of the threat modeling aspect in the three kinds of security research.

5.1 Defensive research

The authors of [38] have examined the security implications of network communication from IoT devices and argued it should be 'default-off'. They provide a natural language framework for creating communication rules.

They describe their threat model in the background section. They state here that they focus on the network as a common denominator of consumer IoT devices and list six assumptions.

1. All parties except the intended users of a system are untrustworthy.
2. Cloud services may become untrustworthy in time.
3. Devices may become compromised and used to launch attacks.
4. Gateways to enforce policies are trusted.
5. Devices, apps and remote servers communicate exclusively through gateways, no side channels.
6. Policing does not create new vulnerabilities to otherwise unmodified systems.

5.2 Offensive research on the protocol level

As a quick summary, the authors of [47] have examined SmartCfg implementations to provision devices without interactive user interfaces. They conclude that bad implementations of the protocol pose a significant security threat.

The interpretation of a threat model in this paper is to make assumptions about the attacker: She is outside the WLAN but can sniff data without any knowledge of which device may currently be configured. Despite not knowing any specifics, the attacker is presumed to constantly monitor the wireless network and thus to be able to observe any SmartCfg event.

Directly, after the threat model another section explains the two main challenges of the attacker. The first is to identify the SmartCfg solution in use. The second is to recover the used provisioning protocol.

5.3 Offensive research on the device level

The authors of [39] made a survey of the security for five different smart lock products. They decided to focus on three kinds of attacks and how these could be exploited in the various products.

Their threat model is in the security analysis section. Aside from the network-based attackers, they devise four kinds of attackers that are deemed important in the setting of smart locks:

1. A physically present attacker can observe legitimate interaction with the lock and can interact with it at any time. She does not posses authorization for the device.
2. A revoked attacker still has legitimate access that is about to be revoked.
3. A thief steals the authorized device.
4. A relay attacker possesses a Bluetooth-enabled device and so also does their accomplice. Neither of them have legitimate access, but one can forward real-time data to the other while at the lock.

They focus their efforts on the four kinds of attackers listed above.

5.4 Summary

While threat modeling has become common practice in IoT security research on the way it is presented still seems to be relatively immature. The authors never reveal the basis for their threat models and do not use common categories that would allow for easy comparison of works in similar parts of the IoT field.

Furthermore, the threat models are usually presented in a way that only focuses on the research topic of the paper without explaining what is deemed irrelevant for the work presented.

This shows that even if we have mature threat modeling frameworks, they are simply not used in the research context. Now that we have established all parts relevant to the discussion: threat modeling frameworks in Sect. 2, the definitions of IoT in Sect. 3 and IoT security research in Sect. 4, we now move on to the discussion of the gaps we claim to have identified between these three areas of research and their impact on research outcomes.

6 Identifying the gaps

So far, we have introduced the three components necessary for making a threat model: A framework to identify the threats, a definition of the field and concrete research questions inside the field.

In this section, we are going to explain the gaps identified between these three components, how the gaps were devised and why they are important.

6.1 The gap analysis

A gap analysis is about the question of where we are at a given point in time and where we should ideally be. Concerning threat modeling and IoT security research, we see three dimensions where there are gaps:

1. A gap between threat modeling frameworks and IoT
2. A gap between threat modeling frameworks and security research
3. A gap between security research and IoT

Before explaining these gaps in detail, we first show the evidence that explains these three gaps. We have reached these conclusions through our literature survey, the results of which were shown in the previous sections. We regard the works we have examined as representative of how the three fields interact. There may be some individual attempts at making the fields interact in a better way, but we will show that in general the fields do not integrate in a proper manner.

Threat modeling frameworks as introduced in Sect. 2 were designed with software in mind. The best example is the Security Development Lifecycle introduced by Microsoft [12]. There is literature available that goes step-by-step through all parts of the software development process and explains the potential pitfalls. We claim that the comparable literature should also guide the development process of IoT devices. The STRIDE threat model is an integral part of this process. It provides a list of threats for each type of software component and hints on how to mitigate them. Of course, the software engineers still have to adapt the abstract suggestions to the project at hand. The process is also based on standard artifacts like UML diagrams that should be available in an orderly software development process. This marks the first major gap since IoT devices cannot be regarded simply as software development projects with some hardware around them, but need to have both sides accounted for equally. This is especially important since usually there is specialized hardware the code runs on as opposed to standard hardware that common software executed on, and this is frequently further abstracted by an operating system.

Between threat modeling frameworks and security research, the gap is in the lack of standardization. As introduced in Sect. 5, the term threat model is often used to name a section that gives a background to the envisioned attacker. However, there are real threat modeling frameworks available that would allow making the research highly comparable and easy to categorize if they were used. When only examining software-based security flaws, as is often the case in IoT security research, it would be feasible to categorize the intended attacks according to STRIDE. Thereby, any reader would immediately understand the impact on the given software.

The gap between security research and IoT is deeply rooted in the origins of IoT security research. As explained in Sect. 3, IoT devices were first regarded as WSN devices and later extended into multi-layered models. Yet, their functionality in the real world is what adds the most value to the devices, because otherwise they would not be functional. It would thus be proper to consider this important base whenever a certain type of IoT device is studied. It is certainly fascinating to consider cases where an IoT smart lock is hacked due to bad access permissions, for example, but all of that should only come after establishing that the smart lock cannot simply be ripped off without the device at least triggering some sort of alarm if the door was forced open. Similar considerations should be kept in mind when studying any IoT device.

In the following subsections, these gaps will be discussed in greater detail.

6.2 The gap between threat modeling frameworks and IoT

The threat modeling frameworks introduced in Sect. 2 are derived from the field of software development. This means that the frameworks care little about physical devices and how they can be interacted with in such a way that the software is not touched. There is the assumption that software is running somewhere and needs to be defended against digital threats that come mainly from a computer network or coding errors in the software itself. The fact that the ‘somewhere’, in other words the device, might be more fragile than the software running on it is not the primary concern.

On the other hand, we have listed three common definitions of IoT. They are of different ages and detail levels, but they have one thing in common: The basis is always some physical device. They may be called WSN or edge nodes but in the end they are devices that are placed somewhere in the real world and can be interacted with by both authorized and unauthorized persons. They can also be interacted with when there is no interface to do so, i.e., by removing them (by force) from the spot where they are meant to be located.

Of course, there is wiggle room in the frameworks to include physical concepts. Let us stay with the example of forceful device removal:

In STRIDE, such an event could be classified as Denial of Service. But this event can only be identified, if the underlying IoT device was equipped with some kind of sensor to assert that this is the case. Otherwise, a sudden outage of one sensor might well be an empty battery and thus not a threat.

Since quantitative threat modeling builds on STRIDE, there is nothing to discuss further about this approach in this context.

In LINDDUN, the focus is on privacy. Thus, we did not find a category in which to confidently place this attack.

Both STRIDE and LINDDUN also rely on data-flow diagrams. This means that there must already have been a phase in the device conceptualization that allows for a data flow to be present. Otherwise, both frameworks are completely blind to this realistic threat.

CORAS relies on UML diagrams and dialogue. While the former would help little in identify the threat, at least the meetings could help in working out the problem and reconceptualizing the product.

To sum it up: Since threat modeling frameworks in the field of computer sciences commonly build on artifacts derived during the creation of software, it is hard or even impossible to identify threats that are solely based on the physical interaction between an attacker and the IoT device. Classic attacks that focus on the software alone can still be modeled with existing frameworks.

6.3 The gap between threat modeling frameworks and security research

We have introduced several papers as examples for different kinds of IoT security research in Sect. 4. Many of these include a description of the underlying threat model. Yet, there is no description of how the various threats mentioned in the given models were derived. The authors claim the threats exist without fully explaining why they are relevant in the given context or why other threats are completely ignored.

Attack scenarios are enumerated that fit the purpose of the paper. This is sensible since it gives the reader a context but an explanation of how the threats relate to privacy or security concerns in a standardized manner is lacking. Thus, they are usually simply left as claims without proper citations or scientific reasoning. If these papers were to refer to threat categories according to a threat modeling framework, it would be easier to compare the research and understand the reasoning of the authors.

6.4 The gap between security research and IoT

While it is clear that IT security research focuses on topics that revolve around software, networking and computer hardware, it seems that in the context of IoT there is an area left untouched that is highly relevant to IoT security. This area is that of the physical aspects of the IoT device under scrutiny. Looking back at the examples from Sect. 4.2, the authors examined several very interesting attacks. However, their security analysis mainly focused on the cloud or network interaction of the devices. Yet, there are far simpler attacks that could take place in the case of smart locks, as an example: How does a smart lock react, if someone prevented it from closing properly? Does it have the capabilities to send an urgent alert? Is there an alarm of some kind built into the lock that attracts attention from people around it?

One may argue that the attacks we have just mentioned have nothing to do with IT security, but IoT is not just about IT as the reference models from Sect. 3 suggest. Furthermore, the question as to whether physical attacks have also been considered in software are just as important as problems with, i.e., WLAN authentication.

7 Conclusion

In this work, we have given a brief introduction to the threat modeling frameworks: STRIDE, LINDDUN, CORAS and quantitative threat modeling as a derivative of STRIDE. We have further outlined three definitions of what makes an IoT device and enumerated common attacks referred to in the literature. Based on these definitions and sample papers from different types of security research, we have argued that there are three gaps in security research for IoT devices.

The first is between threat modeling frameworks and IoT. These frameworks were developed during a time when software could be examined without having too much concern about the hardware it runs on. This is demonstrated by them being based on artifacts from the software development process like data-flow diagrams or UML diagrams. Unless someone has already thought about certain possible attacks in previous software design phases, these frameworks are blind to hardware-based threats. To close this gap, we propose the extending of current threat modeling frameworks to also include artifacts from the hardware design process. It is important to devise ways of assuring that critical device events are passed on to the software level and handled there.

The second gap is between threat modeling frameworks and common security research. Here, we have argued that despite frameworks being available, 'threat models' tend to be explained in just a few sentences the authors come up with. There is no definition of what other problems may exist and why they are not relevant for a given research question. In order to make threat modeling more standardized for security researchers, it is necessary to provide threat modeling frameworks which allow researchers to work with finished products when they have no access to UML diagrams, etc.

The third gap is between security research and IoT itself. Despite reference models for IoT suggesting that there is a physical part, IT security research focuses on applying network attacks or software attacks to IoT but leaves out the new attack concepts these devices bring with them. IoT security research needs to acknowledge that IoT devices are not just network devices with some convenience features in the real world but that it is the other way around. Therefore, research should focus more on these aspects of devices to provide a holistic picture of the threat landscape.

Acknowledgements Open access funding provided by Graz University of Technology.

Funding This study was funded by Technische Universität Graz.

Compliance with ethical standards

Conflict of interest Peter Aufner declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Hussain, S., Kamal, A., Ahmad, S., Rasool, G., Iqbal, S.: Threat modelling methodologies: a survey. Sci. Int. **26**(4), 1607 (2014)
2. Möckel, C., Abdallah, A.E.: 2010 6th International Conference on Information Assurance and Security, IAS 2010, pp. 149–154 (2010). https://doi.org/10.1109/ISIAS.2010.5604049
3. Scott-Hayward, S., O'Callaghan, G., Sezer, S.: SDN4FNS 2013—2013 Workshop on Software Defined Networks for Future Networks and Services (2013). https://doi.org/10.1109/SDN4FNS.2013.6702553
4. Ryoo, J., Kim, S., Cho, J., Kim, H., Tjoa, S., Derobertis, C.V.: 2017 International Conference on Software Security and Assurance (ICSSA), pp. 13–19 (2017). https://doi.org/10.1109/ICSSA.2017.28
5. Vasilevskaya, M., Nadjm-Tehrani, S.: Model-Based Security Risk Analysis for Networked Embedded Systems. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8985, p. 381 (2016). https://doi.org/10.1007/978-3-319-31664-2_39
6. Bhuyan, M.H., Azad, N.A., Meng, W., Jensen, C.D.: Analyzing the Communication Security Between Smartphones and IoT Based on CORAS, vol. 9955. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-46298-1. http://link.springer.com/10.1007/978-3-319-46298-1
7. Nuseibeh, B., Perera, C., McCormick, C., Price, B.A., Bandara, A.K.: Privacy-by-Design Framework for Assessing Internet of Things Applications and Platforms. pp. 83–92 (2016). https://doi.org/10.1145/2991561.2991566
8. Neureiter, C., Eibl, G., Veichtlbauer, A., Engel, D.: Towards a framework for engineering smart-grid-specific privacy requirements. In: IECON Proceedings (Industrial Electronics Conference), vol. 490, p. 4803 (2013). https://doi.org/10.1109/IECON.2013.6699912
9. Alshammari, M., Simpson, A.: Towards a Principled Approach for Engineering Privacy by Design. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), LNCS, vol. 10518, p. 161 (2017). https://doi.org/10.1007/978-3-319-67280-9_9
10. Huth, D.: A pattern catalog for GDPR compliant data protection. In: CEUR Workshop Proceedings, vol. 2027, p. 34 (2017)
11. Johnstone, M.N.: Threat Modelling with Stride and UML. In: Australian Information Security Management Conference (November), vol. 18 (2010). https://doi.org/10.4225/75/57b670493477c
12. Larry, G.: The Security Development Lifecycle: Microsoft (2007). http://download.microsoft.com/download/f/c/7/fc7d048b-b7a5-4add-be2c-baaee38091e3/9780735622142_SecurityDevLifecycle_ch01.pdf
13. Wuyts, K., Scandariato, R., Joosen, W.: LINDDUN: a privacy threat analysis framework (2016)
14. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requir. Eng. **16**(1), 3 (2011). https://doi.org/10.1007/s00766-010-0115-7
15. den Braber, F., Hogganvik, I., Lund, M.S., Stølen, K., Vraalsen, F.: Model-based security analysis in seven steps—a guided tour to the CORAS method. BT Technol. J. **25**(1), 101 (2007). https://doi.org/10.1007/s10550-007-0013-9
16. Luna, J., Suri, N., Krontiris, I.: 7th International Conference on Risks and Security of Internet and Systems, CRiSIS 2012 (2012). https://doi.org/10.1109/CRISIS.2012.6378941
17. Sodiya, A.S., Onashoga, S.A., Oladunjoye, B.A.: Threat modeling using fuzzy logic paradigm. Informing Sci. Int. J. Emerg. Transdiscipl. **4**(1), 53–61 (2007)
18. Peeters, J.: Agile security requirements engineering. Independent, p. 4 (2004)
19. Mohsen Nia, A., Jha, N.K.: A comprehensive study of security of Internet-of-Things. IEEE Trans. Emerg. Top. Comput. **5**(4), 1 (2016). https://doi.org/10.1109/TETC.2016.2606384
20. Ashton, K., et al.: That 'internet of things' thing. RFID J. **22**(7), 97–114 (2009)
21. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput. Syst. **29**(7), 1645 (2013). https://doi.org/10.1016/j.future.2013.01.010
22. Fleisch, E., Weinberger, M., Wortmann, F.: Business Models for the Internet of Things, pp. 1–18. Bosch IoT Lab, Zurich (2014). https://doi.org/10.1007/978-3-642-19157-2_10
23. Green, J.: CTO Data Virtualization: IoT Reference Model Whitepaper (2014)
24. Sen, J.: A survey on Wireless Sensor Network security. Comput. Netw. **1**(2), 55 (2009). https://doi.org/10.1016/j.comnet.2008.04.002. arXiv:1011.1529
25. Andrea, I., Chrysostomou, C., Hadjichristofi, G.: Internet of things: Security vulnerabilities and challenges. In: 2015 IEEE Symposium on Computers and Communication (ISCC), pp. 180–187. IEEE (2015)
26. Borgohain, T., Kumar, U., Sanyal, S.: Survey of Security and Privacy Issues of Internet of Things, p. 7 (2015). https://doi.org/10.5120/15764-4454. arXiv:1501.02211
27. Airehrour, D., Gutierrez, J., Ray, S.K.: Secure routing for internet of things: a survey. J. Netw. Comput. Appl. **66**, 198 (2016). https://doi.org/10.1016/j.jnca.2016.03.006

28. Nawir, M., Lynn, O.B.: Internet of Things (IoT): Taxonomy of Security Attacks. IEEE Conference, pp. 321–326 (2016). https://doi.org/10.1109/ICED.2016.7804660
29. Akram, H., Konstantas, D., Mahyoub, M.: A Comprehensive IoT attacks survey based on a building-blocked reference model. Int. J. Adv. Comput. Sci. Appl. (2018). https://doi.org/10.14569/IJACSA.2018.090349
30. Wood, A.D., Stankovic, A.: Denial of Service in sensor networks. IEEE Comput. **35**(10), 54 (2002). https://doi.org/10.1109/MC.2002.1039518
31. Karlof, C., Wagner, D.: Secure routing in Wireless Sensor Networks: attacks and countermeasures. Ad Hoc Netw. **1**(2–3), 293 (2003). https://doi.org/10.1016/S1570-8705(03)00008-8
32. Douceur, J.R.: The Sybil Attack, pp. 251–260 (2002). https://doi.org/10.1007/3-540-45748-8_24
33. Newsome, J., Shi, E., Song, D., Perrig, A.: Proceedings of the Third International Symposium on Information Processing in Sensor Networks—IPSN'04, p. 259 (2004). https://doi.org/10.1145/984622.984660
34. Anthi, E., Williams, L., Burnap, P.: An Adaptive Intrusion Detection for the Internet of Things Pulse: An Adaptive Intrusion Detection for the Internet of Things (May), p. 1 (2018). https://doi.org/10.1049/cp.2018.0035
35. Ramezanifarkhani, T., Teymoori, P.: Securing the Internet of Things with Recursive InterNetwork Architecture (RINA) (2018)
36. Alrawais, A., Alhothaily, A., Hu, C., Cheng, X.: Fog computing for the Internet of Things: security and privacy issues. IEEE Internet Comput. **21**(2), 34 (2017). https://doi.org/10.1109/MIC.2017.37
37. Yu, T., Sekar, V., Seshan, S., Agarwal, Y., Xu, C.: Proceedings of the 14th ACM Workshop on Hot Topics in Networks—HotNets-XIV, pp. 1–7 (2015). https://doi.org/10.1145/2834050.2834095
38. Hong, J., Levy, A., Riliskis, L., Levis, P.: Don't Talk Unless I Say So! Securing the Internet of Things with Default-Off Networking (2018). https://doi.org/10.1109/IoTDI.2018.00021
39. Ho, G., Leung, D., Mishra, P., Hosseini, A., Song, D., Wagner, D.: Proceedings of the 11th Smart locks: Lessons for Securing Commodity Internet of Things Devices, pp. 461–472 (2016). https://doi.org/10.1145/2897845.2897886
40. Khan, M.A., Salah, K.: IoT security: review, blockchain solutions, and open challenges. Future Gener. Comput. Syst. **82**, 395 (2018). https://doi.org/10.1016/j.future.2017.11.022
41. Hossain, M., Fotouhi, M., Hasan, R.: Towards an Analysis of Security Issues , Challenges , and Open Problems in the Internet of Things. 2015 IEEE World Congress on Services, pp. 1–8 (2015). https://doi.org/10.1109/SERVICES.2015.12
42. Bertino, E., Islam, N.: Botnets and Internet of Things security. Computer **50**(2), 76 (2017). https://doi.org/10.1109/MC.2017.62
43. Duc, A.N., Jabangwe, R., Paul, P., Abrahamsson, P.: Proceedings of the XP2017 Scientific Workshops on XP '17 (7491), p. 1 (2017). https://doi.org/10.1145/3120459.3120471
44. Jing, Q., Vasilakos, A.V., Wan, J., Lu, J., Qiu, D.: Security of the Internet of Things: perspectives and challenges. Wirel. Netw. **20**(8), 2481 (2014). https://doi.org/10.1007/s11276-014-0761-7
45. Apthorpe, N., Reisman, D., Feamster, N.: A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic (2017). https://doi.org/10.14722/ndss.2016.23xxx. arXiv:1705.06805
46. Wood, D., Apthorpe, N., Feamster, N.: Cleartext Data Transmissions in Consumer IoT Medical Devices (2018). https://doi.org/10.1145/3139937.3139939. http://arxiv.org/abs/1803.10147%0Adx.doi.org/10.1145/3139937.3139939
47. Li, C., Cai, Q., Li, J., Liu, H., Zhang, Y., Gu, D., Yu, Y.: Passwords in the Air: Harvesting Wi-Fi Credentials from SmartCfg Provisioning, vol. 11 (2018). https://doi.org/10.1145/3212480.3212496
48. Fuller, M., Jenkins, M.: Security Analysis of the August Smart Lock, p. 16 (2017). https://courses.csail.mit.edu/6.857/2017/project/3.pdf

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.