future internet

MDPI

Article

Segmentation and Filtering Are Still the Gold Standard for Privacy in IoT—An In-Depth STRIDE and LINDDUN Analysis of Smart Homes

Henrich C. Pöhls [1,*,†], Fabian Kügler [1,†], Emiliia Geloczi [2,†] and Felix Klement [2,†]

[1] Chair of IT-Security, University of Passau, 94032 Passau, Germany
[2] Chair of Computer Engineering, University of Passau, 94032 Passau, Germany; emiliia.geloczi@uni-passau.de (E.G.); felix.klement@uni-passau.de (F.K.)
* Correspondence: hp@sec.uni-passau.de
† These authors contributed equally to this work.

Abstract: Every year, more and more electronic devices are used in households, which certainly leads to an increase in the total number of communications between devices. During communication, a huge amount of information is transmitted, which can be critical or even malicious. To avoid the transmission of unnecessary information, a filtering mechanism can be applied. Filtering is a long-standing method used by network engineers to segregate and thus block unwanted traffic from reaching certain devices. In this work, we show how to apply this to the Internet of Things (IoT) Smart Home domain as it introduces numerous networked devices into our daily lives. To analyse the positive influence of filtering on security and privacy, we offer the results from our in-depth STRIDE and LINDDUN analysis of several Smart Home scenarios before and after the application. To show that filtering can be applied to other IoT domains, we offer a brief glimpse into the domain of smart cars.

Keywords: network security; information flow control; Internet of Things; firewall; filtering; Smart Home; STRIDE; LINDDUN; IoT

Academic Editor: Jose I. Moreno Novella

Received: 20 December 2024
Revised: 17 January 2025
Accepted: 20 January 2025
Published: 10 February 2025

Citation: Pöhls, H.C.; Kügler, F.; Geloczi, E.; Klement, F. Segmentation and Filtering Are Still the Gold Standard for Privacy in IoT—An In-Depth STRIDE and LINDDUN Analysis of Smart Homes. *Future Internet* **2025**, *17*, 77. https://doi.org/10.3390/fi17020077

1. Introduction

The widespread use of various interconnected Internet-of-Things (IoT) devices into people's daily routines leads not only to improvements in the quality and comfort of life but also to threats to personal privacy (the worldwide average number of connected devices was already at 17.1 devices per home in 2022 [1]) [2,3]. The massive amount of information generated, transmitted, and stored during the operation of any electronic device may contain data that, under certain conditions, can be used against the user or owner by violating their privacy (even when communication is encrypted, an attack on privacy can work without knowledge of the communication's contents; this type of attack is called *traffic analysis* [4,5], and its countermeasure would be to provide *unobservable communication* for the IoT [6,7]), which may allow attackers to undermine the security of the person [8–10]. Consider two scenarios. In the first scenario, we have a Smart Home that consists of many IoT devices. At first glance, it may seem that the information that is transferred between the devices is not critical. Indeed, data about how often a light or a vacuum cleaner is switched on, without context and separate from each other, in the possession of an attacker are most likely to give them no advantage to perform attacks. However, if communication is observed over a period of time, traffic analysis [4] enables an attacker to identify patterns.

Finally, it allows them to assume that no one is home when messages from the robot vacuum cleaner are observed, while no commands from the lights are sent over the network. Using this information from IoT devices, they can identify with a high probability the time interval when they can gain unauthorized entry to the residence without being detected by the homeowner (more obvious are power or temperature readings from inside the home to identify the homeowner's presence [11]). In the scenario described, the homeowner is probably unharmed but may lose their belongings. The second scenario, on the contrary, has a more pronounced attack vector directed against a person's life and health. Here, we consider a vehicle to which a dongle can be connected. Since the dongle host and the car can bidirectionally communicate with each other, a communication attack, such as a man in the middle, can result in fake sensor information being sent inside a smart car, whereby the car is tricked to 'see' an obstacle in front and performs emergency braking. This can lead to injuries. From a privacy perspective, the smart car's sensor information about speed and velocity allows the determination of driving skills and habits; thus, if leaked to the car insurance and deemed unsuitable or dangerous, this might lead to higher insurance premiums [12].

Among many different approaches to improving the security of the system, in this paper, we focus on filtering, which can enhance the security of the system without significant overloading of resources and limiting the functionality. Filtering is a long-standing method used by network engineers to separate or block different types of traffic from reaching certain devices. In order to evaluate the influence of the filtering on potential attack vectors and their impact on the IoT Smart Home domain, we perform comparative STRIDE and LINDDUN analyses of four common scenarios within the Smart Home domain.

1.1. Contributions of This Paper

The contributions are twofold; first—as a solid basis—we provide full LINDDUN and STRIDE analyses for four generalised Smart Home communication scenarios. To the best of the authors' knowledge, this has not been published in this depth and detail before. Second, we analyse the extent to which various attacks could be mitigated if network traffic filtering were implemented to reduce their impact. We see the role of filtering as twofold, either as an active defensive control (employing a need-to-communicate approach that limits the ability of communications a priori) or as a reactive control (isolating a device following an incident or known vulnerability).

1.2. Organisation of This Paper

In Section 2, we provide some relevant background information. Section 3 discusses works related to our research. Filtering as an approach, especially for increased privacy, is presented in Section 4. Four common IoT use case scenarios and their resulting communication patterns are described in Section 5. Section 7 contains the analysis and summarized results of the impact that filtering has on the threats identified according to the STRIDE and LINDDUN techniques presented in Section 6. Finally, Section 8 concludes the paper and presents the possible future directions of research.

2. Background

In this section, we briefly introduce the terms and methods used in our research. It includes subsections about what we mean by a Smart Home Device, background on the security analysis and threat modelling methods (STRIDE and LINDDUN), and, finally, filtering of network packets as a general concept.

2.1. Smart Home Device

While some authors directly define what is meant by a Smart Home (e.g., [13,14]), we are more focused on the concept of a Smart Home (IoT) Device. Even if the term Smart Home Device is often used without a proper explanation (e.g., [15–17]), there is no single definition that one can refer to. Instead, the term has been defined in various ways with sometimes subtle but crucial differences. For example, Apthorpe et al. [18] define it as 'any single-purpose Internet-connected device intended for home use or a hub-like device that connects and controls multiple single-purpose devices' Apthorpe et al. [18]. This definition suggests that all devices must have some form of Internet connectivity in order to be considered smart; however, there also exist appliances that can operate when limited to the local network. In addition, ref. [16] shows that there are devices that, despite not having Internet, still show limited smart functionality. In [19], the author compares multiple definitions of Smart Home before presenting his own definition of a Smart Home Device as follows: 'A Smart Home Device is a thing, whose main functionality is extended with networking abilities to create a new one. The additional infrastructure for those devices, like a base or control station, also falls in Smart Home' [19]. This definition only requires 'networking abilities' [19], which also include the aforementioned devices that do not rely on an Internet connection. One aspect that both definitions agree on is that also additional devices like control stations, which may not have a perceptible impact on the home, should be considered as part of this environment. This is crucial as in practice the connectivity and smartness of a product may be shared between multiple appliances. (For instance, the Philips Hue Light Bulb can only be controlled with an appropriate bridge. Since 2019, there also exists a version that can be controlled via Bluetooth. In this paper, we consider the version without Bluetooth by default.) Overall, the latter definition offers a good starting point but also has some downsides that should be addressed in a custom definition. One thing that is not mentioned explicitly is that the things should be 'intended for home use' Apthorpe et al. [18] as the first definition points out. Just recently, the international standards community has decided in ISO/IEC 27403 [20] to call this *IoT-domotics* and devote special attention as it requires 'user-friendly interface and usability' as well as being usable by the home users. Extending the general cybersecurity of IoT standard ISO/IEC 27400 [21], it recognises that '[i]n comparison with other IoT solutions, IoT-domotics have specific features and concerns. It is therefore essential to adapt the general IoT security and privacy principles to IoT-domotics and provide stakeholders with thorough and tailored guidelines in specific scenarios of IoT-domotics.' [20]. Finally, there exist applications that have not been possible with traditional devices at all, and therefore, there is no such thing that can be extended. For example, even if Amazon referred to Amazon Echo as a 'Smart speaker' (https://www.amazon.com/gp/product/B07NFTVP7P, accessed on 19 December 2024), it can also be seen as primarily being a voice assistant.

Based on previously mentioned definitions, we identify our own definition of a Smart Home Device as follows:

- A *Smart Home Device* is a thing that is intended for home use whose functionality is only possible or whose main functionality is extended through means of networking capabilities including additional infrastructure contributing to this.
- A *Smart Home* is a system incorporating Smart Home Devices.

Similar to Smart Home Devices, there also does not exist a single accepted definition for either the IoT or Cyber-Physical Systems (CPSs), and a distinction between the latter might often not even be clear. An extensive collection of definitions for both IoT and CPSs can be found in [22]. This in fact makes it hard to make precise statements about the relation of these concepts. One clear distinction is that both IoT and CPSs are not limited to the home area but are seen in a broader context.

IoT devices, for example, exist in many areas, including healthcare, infrastructure, and many others [23]. Conversely, however, as the name suggests, for a thing to be an IoT thing, Internet connectivity can be seen as a mandatory characteristic. While some authors assume the same for Smart Home Devices, this does not necessarily apply to all of them, as discussed previously. Therefore, neither is Smart Home a sub-concept of IoT nor vice versa. However, as a lot of devices in a Smart Home are indeed IoT things, research on IoT can often be applied to this domain as well.

As for CPSs, one can come to the same conclusion when assuming their close relation to IoT or treating them as indeed a more generic concept compared to Smart Homes when the Internet is not mandatory. In any case, similar to the IoT, research on CPSs might be relevant for the Smart Home domain as well.

2.1.1. Device Types

A Smart Home typically contains Smart Home Devices (SHDs) as defined before, as well as general-purpose computing devices (GPCDs) like personal computers (PCs) or Smartphones, both of which communicate with other devices in the local network or the Internet. Additionally, it should be noted that it is not always straightforward to distinguish between those groups. For example, some software like Home Assistant (Version: 2025.1.3 https://www.home-assistant.io/ (last accessed: 19 December 2024)) can turn a computer into a Smart Home control centre, which would fall into the SHD category if this is the sole objective of the device. If instead it also contains other applications, such as a web or file server it may be more appropriate to assign it as a general-purpose computing device.

A major difference between SHDs and GPCDs lies in the diversity of their communication with other devices or servers. A PC or Smartphone might access any website on the web, download files from any server, or send e-mails, to only name a few scenarios. In addition, such computers also connect to various devices in the local network, e.g., to sync with a local file server, print some files, or control the Smart Home Devices. SHDs, on the other hand, are usually limited to a small set of servers they connect to [24]. For example, Notra et al.'s investigation shows that the Nest Smoke Alert only connects to four domains on a daily basis and only three of them in the case of an emergency [25]. While such SHDs with relatively low customizability also have an accordingly static set of domains they connect to, those that can be customised by the owner, e.g., by means of an add-on or apps, can have a more dynamic network behaviour. For example, the communication of the Amazon Echo depends on the 'skills' that are installed for Alexa [26].

2.1.2. Communication Patterns and Smart Device Connectivity

Traditional computers are typically connected to a local network directly over Ethernet or WLAN and are able to communicate with other devices in the network over a switch, which is typically incorporated into the home router, which often also contains an access point for the WLAN. In contrast, the Smart Home domain incorporates a broader spectrum of communication methods. While there are many products that follow the "traditional" model, there are also devices that use different communication ways either additionally or exclusively. Even for the same use case, different approaches can be applied. While, for example, Yeelight bulbs are directly connected to the WLAN ("Yeelight smart LED products support remote control through WiFi"; see specification found at https://www.yeelight.com/download/Yeelight_Inter-Operation_Spec.pdf (last accessed: 19 December 2024)), Philips Hue light bulbs are not connected to the local network directly but through an additional bridge ("anything "smart" needs to be able to receive instructions. Hue smart lights use Zigbee or Bluetooth to communicate, depending on whether you have a Bridge," as stated on the explanation found at https://www.philips-hue.com/en-us/explore-hue/how-it-works (last accessed: 19 December 2024)). The ZigBee protocol is used for communication between the bridge and the actual lamp [25]. As traffic on other communication technologies behind such a bridge cannot, however, be observed by a firewall operating on the IP traffic in the WLAN or Ethernet of the home network, it is out of the scope of this paper. Also excluded is any other form of communication that does not involve the local area network (LAN)—be it wireless (WLAN) or cabled—such as Bluetooth connections between a Smartphone and smart device, direct WLAN communication between a Smartphone and the WLAN access point of the smart device, or communication completely outside of the home, e.g., among several servers on the Internet.

Even when focusing on such communication that is visible in the local network, there exist generalisable and fundamentally different communication patterns in a Smart Home. Table 1 gives an overview of patterns, which involve only two communicating parties, in order to illustrate the diversity of control or information flow in the Smart Home domain. In addition, there exist control flows or information flows that involve multiple of the above patterns. Examples of such indirect control communication patterns involving three parties are given in Table 2. Of course, even more arbitrary complex patterns could arise. Finally, it should be noticed that those patterns are not mutually exclusive but one device can use multiple of them.

Table 1. Communication patterns involving two parties.

Initiator	Target	Example
GPCD	GPCD	PC in the local network syncs files with the local file server.
	SHD	Smartphones control the lights with an application.
	Internet	PC accesses a web page.
SHD	GPCD	Motion sensor sends a status update to Smartphones.
	SHD	Smart hub controls a device directly in the local network.
	Internet	Smart thermostat uploads temperature profile to the cloud.
Internet	GPCD	PC outside the network syncs files with the local file server.
	SHD	Manufacturer sends updates to the device.

Table 2. Communication patterns involving three parties.

Initiator	Relay	Target	Example
GPCD	SHD	SHD	Smartphones controls lights indirectly over a smart hub.
		Internet	Smartphones control devices that sync their state with the cloud.
	Internet	SHD	Smartphones controls device indirectly over a server.
SHD	Internet	SHD	A smart hub controls devices indirectly over a server.

2.2. Security Analysis and Threat Modelling

We aim to obtain a provably secure design that benefits from model-driven security approaches, as they allow us to identify and mitigate threats in the early phases of development for cyber–physical systems [27] or the Smart Home.

According to the work by Shostack [28] in 2014, threat modelling is a process of finding threats to a system. The author proposes a framework of threat modelling consisting of the following steps:

1. Modelling the system itself,
2. Finding threats to the afore-modelled system,

3. Addressing the threats,
4. Validation.

Shostack defines three main focuses of threat modelling: asset-centric, attacker-centric, and system-centric [28]. Following the suggestion from the original work we will use the latter:

System-centric approaches try to elicit threats based on a model of the system itself. Focusing on the system, in contrast to the aforementioned alternatives, has the benefit of being able to perform abstract analysis and does not require thinking about the different types of attackers.

2.2.1. STRIDE Threat Model

One of the most common threat modelling techniques is the STRIDE method, which was developed by Kohnfelder and Garg in 1999 at Microsoft [29]. In this method, threats to the system are considered in six categories, each violating one of the properties necessary for the secure functioning of the system [30]:

- **S**poofing refers to an attempt by an attacker to impersonate an approved user or system element and use their identity to access information or services. Therefore, secure system *authentication* may be compromised.
- **T**ampering relates to the malicious alteration or deletion of data, which violates the *integrity* property of the system.
- **R**epudiation means an attacker is able to deny their actions, which leads to a violation of the *non-repudiation* of the system.
- **I**nformation Disclosure involves unauthorised disclosure of information, compromising *confidentiality*.
- **D**enial of Service threat utilisation violates the *availability* of the system for approved users.
- **E**levation of Privilege refers to granting users access to services, information, or system elements to which they should not have access due to their role restrictions. During the exploitation of this threat, the secure *authorisation* of the system is compromised.

The application of STRIDE consists of the following five steps, as also depicted in Figure 1.

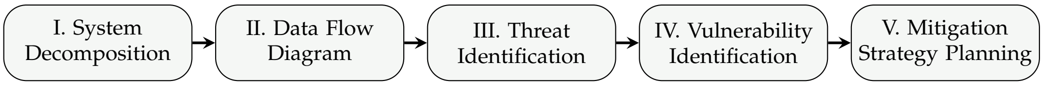
I. System Decomposition
II. Data Flow Diagram
III. Threat Identification
IV. Vulnerability Identification
V. Mitigation Strategy Planning

Figure 1. STRIDE Steps.

I. *System Decomposition*. To begin with, it is necessary to split the analysed system into logical or structural components. It can be processes occurring in the system or in which the system participates, physical elements of the system, or ones that communicate with the system.

II. *Data Flow Diagram (DFD)*. Next, for each system component defined in Step I, it is necessary to create a DFD that reflects its functionality. Some of the basic DFD elements include external entities, process, data flows, and data stores.

III. *Threat Identification*. During this step, for each system element, possible STRIDE threats should be identified.

IV. *Vulnerability Identification*. Further, it is required for each previously identified 'per-element-threat' pair to determine the vulnerabilities of elements that can be used to enable the threat.

V. *Mitigation Strategy Planning*. Finally, for each vulnerability, strategies for mitigation should be proposed/developed.

2.2.2. LINDDUN

LINDDUN is a privacy threat modelling framework designed to systematically identify and address privacy risks in IT systems [31]. The name LINDDUN is an acronym derived from the categories of privacy threats it aims to address:

- **L**inkability refers to the risk that two or more pieces of data can be linked to derive additional information about a person.
- **I**dentifiability describes the possibility of uniquely identifying a person from a dataset.
- **N**on-repudiation is the inability of an entity to deny having performed an action.
- **D**etectability refers to the risk that the existence of data or a system can be detected by unauthorised parties.
- **D**isclosure of Information is unauthorised access to private data.
- **U**nawareness relates to a lack of awareness or content about how data are collected or used.
- **N**on-compliance means violation of legal, regulatory, or contractual obligations regarding privacy.

The application of LINDDUN consists of the following six steps, as depicted in Figure 2:

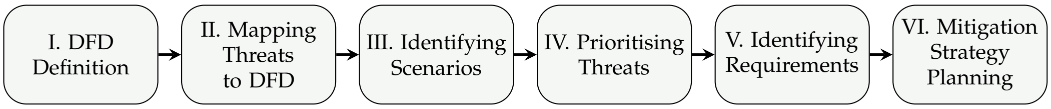
I. DFD Definition
II. Mapping Threats to DFD
III. Identifying Scenarios
IV. Prioritising Threats
V. Identifying Requirements
VI. Mitigation Strategy Planning

Figure 2. LINDDUN Steps.

I. *Data Flow Diagram (DFD) Definition*. To begin with, it is necessary to provide a high-level system description identifying included components, data flows, and processes.
II. *Mapping Threats to DFD*. Next, for each system component defined in Step I, it is necessary to identify related LINDDUN components.
III. *Identifying Scenarios*. During this step, potential threats and usage scenarios for each category should be identified.
IV. *Prioritising Threats*. Further, evaluating the risks based on their likelihood and impact is required.
V. *Identifying Requirements*. During this step, privacy threats and requirements should be mapped.
VI. *Mitigation Strategy Planning*. Finally, strategies to address or minimise the identified privacy risks should be proposed.

2.2.3. STRIDE/LINDDUN Per Element

Around both strategies, techniques were built to better focus on the parts of the system where certain threats are likely to arise, e.g., STRIDE per element or STRIDE per interaction [28] and LINDDUN per element (the authors do not call their approach LINDDUN per element themselves, but as the corresponding approach is an exact equivalent of STRIDE per element for STRIDE, it will be referred to as such for convenience). We used those as our main techniques for finding threats. In both cases, entities are mapped to a subset of the corresponding threats. A mapping for the STRIDE per element approach as used by Microsoft is given in [28]. The mapping for LINDDUN per element is given by the authors themselves in [31].

2.3. Data Flow Filtering Concept

In general, during the transmission of information from a source to a destination, the data containing this information are transmitted over a network, generating network traffic or data streams. In today's usually packet-switched networks, the data stream consists of packets [32]. In this paper, we use the term '*data flow*' to refer to a sequence of packets that are transmitted between two participating devices of a network, such as an IoT device and an Internet server, or two IoT devices within the same local network. Technically, the network infrastructure tasked to transport that data could thus prohibit, e.g., block, such flows. For example, preventing the transmission of information from a particular source may be required. This can be reached by filtering the data flow.

In this paper, we are interested in blocking information by blocking the transport of network packets that contain such information. Figure 3 shows the idea where the connecting lines stand for network connections, e.g., physical cable infrastructure like in a car (Figure 3a) or wireless network connections like in a home network with a WLAN [33] (Figure 3b). We assume that the process carrying out the filtering is able to see, evaluate, and block the traffic according to rules. In a star-shaped network infrastructure, like the WLAN network of a Smart Home, this function can be located at the WLAN access point (see Figure 4); of course, this could also be distributed [34]. Additionally, other means of filtering could be applied that are more centric to the information, e.g., adding random noise to data or pruning anomalies; this is often also rightly termed 'filtering'. For example in larger IoT deployments such as smart office buildings, there is often building energy management systems (BEMS) that would benefit from trustworthy and 'clean' data as inputs and thus undergoes what is called 'data treatment' [35] or 'data filtering' [35] to increase the data quality. Likewise, but quite to the contrary, the idea of privacy preserving filtering of authenticity-protected energy consumption data [36] adds noise to data to decrease the data quality to a level suitable for both the application and the user's privacy preferences. We briefly touch on the impact of filtering on the data quality in Section 2.6. Moreover, filtering data on the level of network protocols on higher layers within the OSI stack [37], e.g., like filtering and transforming the data's value within messages conforming to the MQTT protocol for increasing Smart Home privacy [38] or within the CAN-Bus of a car [39,40]. Note, we are not concerned with this data-centric type of filtering in this paper, but with network-centric filtering and the resulting network segmentation.

This idea of filtering data within network packets transmitted over a network to increase security is not new and was discussed in the very early days of computer networking [41]. The prevailing concept from Bellovin and Cheswick [41] is as follows: a firewall is a device that analyses the network traffic and enforces a set of security policies to block or allow certain data flows. The firewall's goal is to create a barrier between one side of the network, which is deemed to be the inside, and the other side, seen as the outside world. Thus, it prevents unauthorised access to the inside part and thus protects against attacks from the outside.

For example, Figure 3 shows the general idea and a possible path the attacker's network packets can travel (*attack vector*). Corresponding to the scenario, the attacker could try to influence the behaviour of the smart car or an IoT device (indicated in Figure 3a,b by a red explosion), or even fully disable the device (indicated in Figure 3b by a red cross). Thinking of the data flow as a continuous stream of packets that is exchanged between two devices over a network connection [42]. Hence, a network firewall should be placed on the path of that data flow so that it has an opportunity to intercept packets. Figure 4 shows how filtering deploying '*Firewall*' functionality helps to remove the attack vector (i.e., the path a malicious packet could travel) and thus mitigate or remedy the attack's impact.

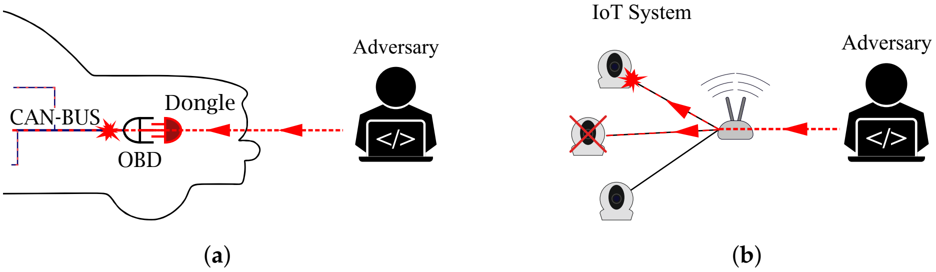

Figure 3. Attack against devices (**a**) inside the car coming from the outside via the On-Board-Diagnosis (OBD), (**b**) inside the IoT system coming from outside via the Internet or via an adversarial device connected to the home network.

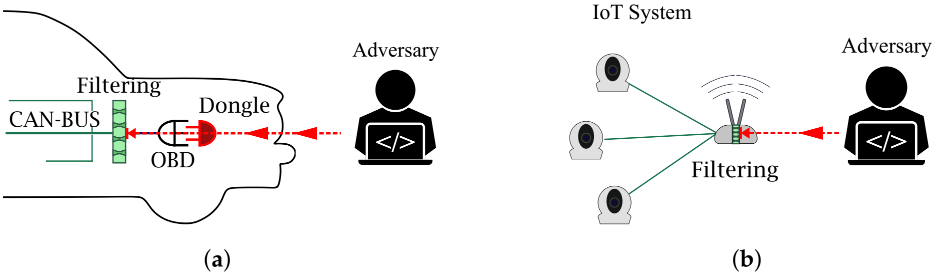

Figure 4. For filtering a firewall is placed between the inside network and its connections to the outside; e.g., (**a**) behind the OBD interface of the car or (**b**) the Internet connection of a router.

2.4. Technical Implementation of the Filtering Approach Using Bark

We have previously described the general approach of filtering (in Section 2.3). Next we describe one technical solution to achieve such filtering, as well as the approach to build the filtering rules.

In [26], Hong et al. present an approach we will refer to as *Bark* in this paper. Their approach offers fine-grained access control policies, not only with respect to remote participants on the Internet, but also between devices in the local network. In contrast to other similar approaches (e.g., [15,17,43], etc.), there is a specification of rules for local network traffic.

While the authors consider the problem of authenticating clients to be out of scope, similar to the well-known work *IoT Sentinel* of Miettinen et al. [17], they also briefly mention the possibility of using different pre-shared keys at the access point. Something we have termed *per-device-802.11-passphrase* and briefly discuss in future work (Section 8). Further, the approach is based on a *default-off* policy, which means that by default no communication is allowed at all. Permitted communication is then whitelisted using a policy language, called *Bark*. *Bark*'s format is inspired by natural language constructs *rules*, containing a *subject*, an *object*, an *action* and *conditions*.

Both *subjects* and *objects* include:

Who—represents the device notion, and corresponds to a MAC address or an IP address, for local or remote devices, respectively. However, in the rules, the *who* is referenced by a human-readable name which is mapped to the corresponding identifier when a device is configured. For convenience, there is also the possibility of grouping different entities.

Where—corresponds to the gateway that a device is connected to. However, because only one gateway is considered here, this aspect is ignored (Note, also the original work [26] does not explain how this would be enforced for multiple gateways).

Only *objects* includes:

What—corresponds to a service or protocol offered by a device (e.g., HTTP, SSH, etc.) and is in most cases simply mapped to a transport protocol (TCP, UDP) and a port but can also be more complicated like DNS requests to only a specified set of addresses.

Only *action* includes:

How—relates to protocol-specific actions (e.g., GET/POST for HTTP) which also shows that the approach considers deep packet inspection when applicable.

Only *conditions* include:

When—specifies time constraints, e.g., operation time.

For example, a Bark rule can be as follows:

```
Allow the Robot Vacuum Cleaner, at router, to connect
https/tls(XXXX) of Vacuum Cleaner's Servers on mondays.
```

Of course, the above rule would require –among other details– the specification of the DNS names or IP addresses of all the allowed *Vacuum Cleaner's Servers*, as well as the MAC address or IP address of the *Robot Vacuum Cleaner*.

We conclude that this approach provides a flexible way to filter not only external but also Local Network Communication. Hence, we decided to use it as a technological basis for our evaluation of the impact that filtering has.

2.5. Creating Rules for Filtering

While the rules, e.g., for Bark as shown in Section 2.4 may be easy to read, Hong et al., as the authors of the original work [26], admit that creating the mappings between human-readable names and the underlying entities (port numbers, sets of permitted IP addresses, domain names, API-specific constructs, etc.) may require knowledge and skills that not every user has. Hence, such mappings could be done by experts and be loaded when a new device is added [26].

The end user could copy and use rules that have 'been defined once by an expert' [26] and published, e.g., by communities of users or the device manufacturers. The end user could also benefit from automated guidance, e.g., in [44] the authors Anselmi et al. introduced a tool named *COPSEC*. COPSEC allows to evaluate whether an IoT device is compliant with security guidelines and privacy regulations, e.g., the European Regulation on Data Protection (GDPR) [45]. If the user would trigger such an automated evaluation for his own device and the result would be negative Bark-style filtering rules could be automatically suggested based on the analysis from COPSEC. Then, once filtering is in place, the user could re-run the COPSEC analysis framework to check if filtering has improved. Also, Intrusion Detection Systems–conventional or those based on artificial intelligence [46]–could also be used to prepare rules and present them to the end user as suggestions. Further, there is research in the human-computer interaction domain suggesting that end users, once empowered, could be more willing to try building rules on their own [47]. This would require a playful and dynamic interaction with visualisations of the data flows to enable and disable those, i.e., building a rule from a simple live or historic data flow visualisation [47].

Moreover, it is assumed that the rules are restrictive enough to only allow the communication that is necessary, as depicted by the Data Flow Diagrams for our Smart Home communication scenarios.

2.6. Impact of Filtering on Performance and Data Quality

The goal of filtering for privacy for this paper is to limit the information that is gathered, thereby reducing the negative impact on the privacy of the Smart Home user(s) to a level deemed tolerable by them. Furthermore, the privacy of all individuals affected by the Smart Home's data acquisition should be considered (We here only talk about the filtering applied by the owner of the WLAN network and the devices joining this network. See works like [48] for an idea on how you could protect the privacy of users even in unknown network) The filtering function introduces an overhead that negatively impacts the performance of the networking infrastructure. This overhead arises from two primary factors: the number of rules that have to be evaluated before a match is found and the complexity of each rule. Although the exact impact cannot be quantified in this context, the study [49] by Lyu and Lau suggests that filtering at a level that does require that 'the firewall analyzes application commands inside data packets and keeps logs' [49] and that such a firewall '... incurs higher overhead than a simple packet filtering firewall ...' [49].

Therefore, we decided to evaluate the impact of packet-level filtering on privacy and security. Figure 5 gives the performance overhead. We implemented our own rule-based firewall with at least the same capabilities as *Bark* (This is not further elaborated in this paper, as measuring the implementation of filtering was deemed beyond its scope. However, the provided analysis aims to provide indicative values for the performance impact of filtering) and as expected it showed tolerable overheads for bandwidth and latency (We measured the bandwidth using the `iperf` tool over a period of 60 s and the average latency using the `mtr` tool over 60 cycles). Overall, 15 configurations were evaluated, with the number of installed network devices ranging from 50 to 700 and varying rule complexities requiring between 50 to 250 checks to evaluate all rules. In an Ethernet-based setup, the results indicate that the performance mainly depends on the number of installed rules, with only a slight increase observed as the number of network devices grows. The latency, measured at a maximum of 3.7 ms for Ethernet with 700 devices and 250 rules can be considered sufficiently low and unlikely to have any noticeable impact, especially in a home network setting. For the WLAN bandwidth, the baseline of the experimental setup was 11.66 Mbit/s, constrained due to the hardware used. In the worst-case scenario with the maximum configuration, the WLAN bandwidth decreased to 10.86 Mbit/s. In our Ethernet setup, the impact on bandwidth becomes clearer: On prototypical hardware (The evaluation was running on a Raspberry Pi 4B with 4 GB of RAM (https://www.raspberrypi.org/products/raspberry-pi-4-model-b/ (last accessed: 19 January 2025)) and Linux Kernel 4.19; the WLAN was served via an external USB WiFi dongle based on the Ralink RT5572 chipset) the bandwidth halves to 463 to 451 Mbit/s already in case of only fewer rules and 50 devices as compared to the baseline values of 915 or 940 Mbit/s. In the maximum test with 250 checks within the rules the value decreases to 76 Mbit/s in our prototypical implementation. However, even that value should be sufficiently high for most communication in the local network and is expected to increase if network filtering optimised hardware like in modern routers would be used or other optimisations, e.g., on the order of rules [50], would be facilitated. However, even that value remains sufficiently high for most communication in a local network. Overall, our comparison of different test configurations shows that performance impact depends mostly on the number of rules, while the number of devices shows a comparatively smaller effect.

Also, the original work [26] indicates that Bark provides its functionality with a reasonably acceptable overhead in most use cases. Their experiments indicate 'that, for a reasonable number of applicable rules (<10), the additional latency is unnoticeable to users' [26]. However, the original work does not include bandwidth measurements.

Overall, the above findings suggest that filtering on the level of network packets (Filtering here operates at *level 3*, also called the *network layer*, of the OSI layer model [37] and allows that filter rules can be based on IP addresses and ports), even with non-optimised rule-parsing and non-optimised rules, e.g., 'latency increases as the number of rules that are statically matched and fruitlessly evaluated increases' [26], can be sufficiently fast.

Successful filtering, i.e., blocking a network packet and its contained data from reaching its destination, can result in reduced data transmission and thus in a reduction of the data quality. Reducing data quality slightly is a common privacy protecting mechanism and can counter problems like that too fine-grained energy values allow detecting the use or mode of use of electrical appliances within a Smart Home [51,52]. Depending on the application's need in terms of data quality, the application's functionality could be able to tolerate a certain amount of blocked packets or totally cease to be useful. For example, assume the application in which a constant video stream of a video camera is sent to an Internet server in order to alert the user when motion is detected and being able to receive a recording of the event that triggered the warning on the user's Smartphone. However, capturing the hallway during normal times could be forbidden by a rule in order to preserve the privacy of the people in the Smart Home, but could be enabled for dates on which the user is away on vacation to be alerted of burglars or other events.

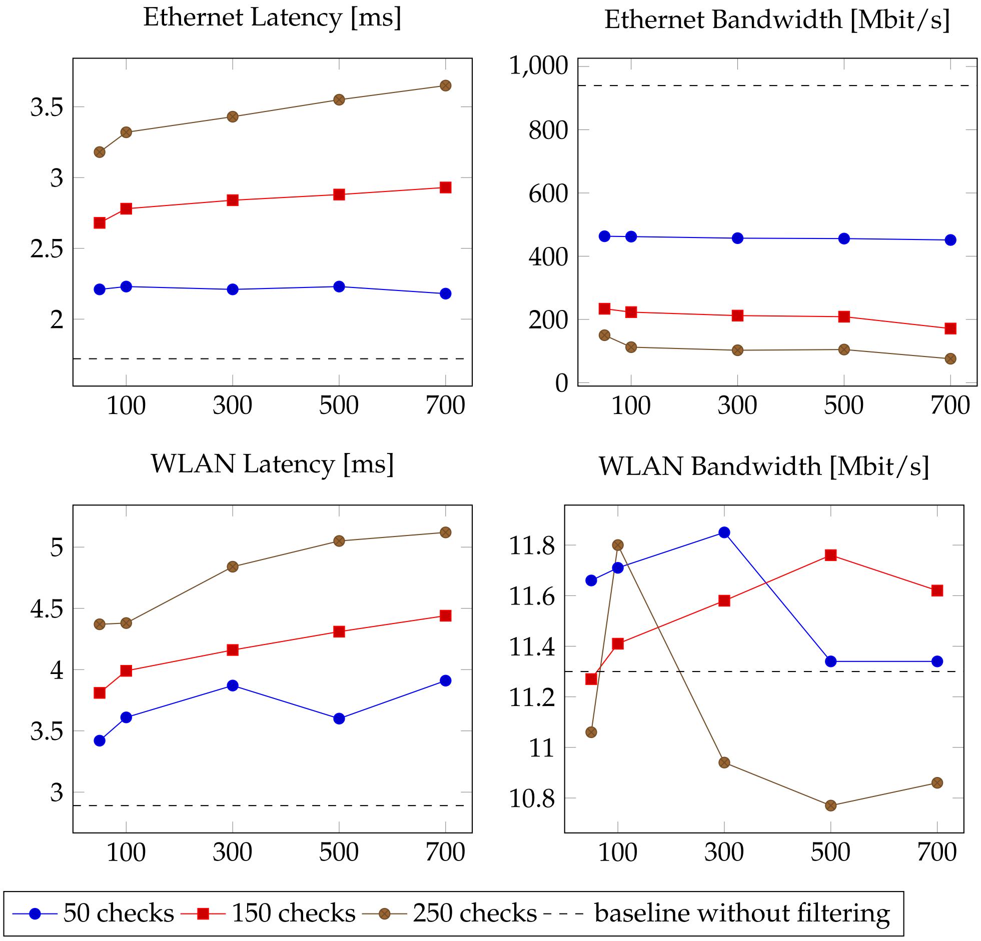

Figure 5. Performance evaluation of our own Bark-style prototypical filtering setup tested on a Raspberry PI 4B. The x axis corresponds to the number of network devices ∈ {50, 100, 300, 500, 700}. Apart from the baseline value, three graphs are shown, representing the number of checks that are needed to evaluate the tested set of filtering rules ∈ {50, 150, 250}.

Overall, filtering can reduce the data quality and it is out of the scope of this paper to discuss it in more detail. In this paper we provide generalised usage scenarios and their dataflow (see Section 5), the impact of filtering on privacy and security has been measured without any compromise of the functionality enabled by the indicated data flows. For example, if the communication with an outside server is in the dataflow for that scenario, then the assumption is that filtering preserves that and hence the functionality should be fully preserved. If one wants to balance privacy concerns and a minimum level of data quality then this would require mechanisms allowing to set maximum limits on the amount of data quality decrease, e.g., comparable to [36]. If–even further–a partial loss of functionality could be tolerated by the user, then a playful and dynamic way to interact with the rules could enable the user to directly see the consequences of a rule on the functionality, e.g., as suggested in [47]. For example, a user that is blocking Internet communication of the smart lights using filtering would notice that this removes remote accessibility to smart lights via an application on a Smartphone when outside the home's WLAN but might notice that it still retains the Smartphone application's functions while the Smartphone is in the local WLAN.

3. Related Work

Although our research focuses on solving privacy problems in Smart Homes through filtering, this topic is quite broad and can be addressed from different perspectives and using different methods. Since it is impossible to mention every work in this domain, in this section, we discuss only a few works that we believe are the most relevant to our study.

In [53,54], a risk analysis model for a Smart Home Automation System called *SHAS* is presented. (The two papers by the same authors likely present results of the same risk analysis. While the results are slightly different, the discrepancy can be explained by the fact that in the former, the risk values were rounded before categorisation.) While the model focuses on ZigBee and Z-Wave devices, which can also be controlled remotely over a dedicated cloud service, the scenarios that were considered throughout this paper consider an IP-based Smart Home, where all sorts of devices can in principle communicate directly with each other. As a result, the authors identify 32 risks with repeating vulnerability and threat descriptions.

Similar work was carried out in [55], where Geneiatakis et al. discuss a handful of rather abstract security and privacy threats, namely Eavesdropping, Denial of Service, Impersonation and Software exploitation, for which they discuss potential attack vectors.

Heartfield et al. present a comprehensive threat taxonomy [56] that focuses on the Smart Home domain. Here, threats are considered along three dimensions, with the first one focusing on different communication protocols, the second on security and privacy impacts on the system, and the last one on the impact on physical symptoms and psychological reactions of the home residents. Together, they provide a wide overview of vulnerabilities that were identified in the literature.

In order to enhance the security and privacy of Smart Home systems and mitigate threats and vulnerabilities discovered in previously mentioned works, numerous different approaches and techniques have been developed. In our work, we discuss a few of them that are closest to our main focus.

In [57], De Donno et al. propose the use of a white worm, called *AntibIoTic*, which invades vulnerable devices and then removes existing malware and closes vulnerabilities such as unnecessary open ports or default passwords.

Additionally, it adds network filtering at a central node, which either grants or completely prohibits Internet access for a device, depending on the configured operation mode of the network as well as depending on the success of invading and securing the device.

In [58], while mainly concerned with the security and privacy issues of Hybrid Broadcasting Broadband TV (HbbTV), Ghiglieri and Waidner also discuss the use of a firewall for Smart Homes in general, emphasising the importance of a usable GUI for the acceptance of such an approach. They notice that technical details such as IP addresses and port numbers should be hidden from the user.

In [59], Gebhardt et al. propose and implement a gateway for Smart Homes that incorporate sensors and actuators that use the KNX building automation standard. The gateway transports information to external endpoints and allows remote control using the Session Initiation Protocol (SIP). During research, the authors noticed the need for different roles, such as guests. They propose using a three-layer security architecture consisting of encryption and two access control lists in order to limit the device and status information access.

Barrera et al. specify a network filtering-based approach, called *IDIoT* [24], which restricts traffic leaving the network based on policies. They discuss three options for obtaining such policies, namely the manufacturer or third parties or by observing network traffic, an idea which has also been proposed by [43,60]. The policies allow specifying network endpoints, communication limits, and application-specific restrictions, e.g., regarding DNS requests and responses. Additionally, while the authors present an example policy and discuss the use of additional options, no real specification is given, while the Rule Packages in this approach are formally specified using the JSON schema, and so is the provider interface. Similar to the approaches in [17,60], they divide the home network into segments for trusted and untrusted devices that might, however, hinder the local device discovery or communication.

In [43], Serror et al. present their idea of restricting network communication inside as well as leaving the local network to the minimum that is needed to preserve the functionality of a device. Similar to [24,25] they also provide an example of rules that could be fetched from an external provider but do not provide a formal definition of the rule format. However, while they emphasise the importance of communication control inside the local network, they miss the opportunity to detail this concept, and the authors also did not implement their approach.

Hong et al. propose novel network filtering-based approach *Bark* [26]. The approach offers fine-grained access control policies, which are formulated using a format that is inspired by natural language constructs. While those rules make it possible to control both remote and local communication according to the user's needs, formulating them can still be too hard for a non-technician. For example, it still requires the user to know what interactions a device really needs in order to work properly.

In addition to considering different approaches to applying filtering to the Smart Home, we also see the need to add filtering mechanisms that are used in smart cars to this section, as they are inherently a special case of IoT devices. Today's cars, self-driving or not, exchange information with the outside at least over a standardised interface: the On-Board Diagnosis interface (ODB-II, or ODB for short). This interface enables reading a car's speed, revolutions per minute, and other data. The USA mandated this interface for all new cars starting in 1996, and EU regulations [61] required it for gasoline cars from 2001 and diesel cars from 2004. For information transfer outside the smart car, a small portable device called a *Dongle*, is usually facilitated and connected using the OBD interface. This information comes from so-called Electronic Control Units (ECUs). An ECU is a computerised system within the vehicle dedicated to a function which communicates via a network called CAN (Controller Area Network), or CAN-BUS [62,63].

Dongles' security is analysed by Wen et al. in [64], in order to provide a comprehensive vulnerability analysis of 77 On-Board-Diagnosis (OBD-II) dongles. In the paper, the authors propose an automated tool called DongleScope to perform an analysis and test the dongles. Moreover, the authors identify and define different types of possible vulnerabilities.

In [65], Yadav et al. give an overview of various security vulnerabilities and points of entry for malicious entities in vehicular systems.

Several works are dedicated to researching information that can be gained from a car and how it can influence privacy [66–69]. All works describe how to monitor cars, predict the condition of the internal hardware, detect driving habits, and discover different anomalies.

Bernardini et al. define and explain eight security requirements and five safety requirements for vehicle systems [70]. They also describe how existing systems and solutions can be used to fulfil these requirements. Furthermore, the authors explain in detail what safety concepts are to be pursued in vehicle systems and possible problems or limitations that may arise.

In [71], Hoppe et al. show the need to examine and modify already existing security vehicle systems. The authors prove that the Intrusion Detection System does not fully protect cars from intruders, even though this system is one of the newer ones in vehicle safety. They conclude that improvements need to be made to minimise the risk of attacks. This shows us that even current security concepts are often not fully developed and cannot offer complete protection.

Studnia et al. discuss fundamental problems related to car security [72]. The results of their research show that the computing power of a car is very limited and that this can lead to problems when using strong cryptography within certain protocols. In addition, they conclude that car manufacturers must validate the software running on an ECU embedded within a vehicle and test it periodically to guarantee its integrity. An entire vehicle can become vulnerable if bugs remain in the vehicle system. These effects are of course reflected in the severity of the respective bug. If a security flaw is exploited, it can require anywhere from several months to years for a patch to be installed for all of the specific cars that were already on the road. This implies that it is an extremely important task to prevent malicious code from entering the vehicle in the first place.

Wolf et al. [73] examine architecture and threats that are prevalent in modern vehicles. They discovered that the gateways built into the automotive network require the use of powerful firewalls. In addition to this, they stated that the firewall implemented in the gateways also needs to possess rules that control access based on the security relevance of the particular network.

While filtering approaches or firewall concepts for networks inside the vehicle do exist and are not completely unknown, the range of available research is very limited, especially compared to works on inter-vehicle networks. Even less information is available about existing solutions for cars being deployed. For example, NXP describes the need to protect the car's networked devices from unwanted outside traffic by a gateway for 'filtering inbound and outbound network traffic based on rules, disallowing data transfers from unauthorized sources'. NXP further states that a more fine-granular approach '[...] may include context-aware filtering' [74]. But, often, the exact mechanisms and the security functions used in real vehicles are not published. Another manufacturer's solution is the 'Central Gateway' for central in-vehicle communication from Bosch, which lists a firewall and an Intrusion Detection System on its product page [75]. However, neither the info PDFs nor the actual page provides more precise details. Even when we specifically asked the responsible department, we were unable to obtain any further information about the security features mentioned. In 2016, the company Karamba Security [76] released a security architecture that acts as a gateway between a car's access points and critical networks/modules. Karamba calls it ECU Endpoint Security. Dropper Detection and Malware Prevention. To define factory policies, the developers had the idea of having a system embedded directly in the firmware. This is to prevent malicious code from infiltrating the system. Each ECU specifies its own policy and generates a so-called whitelist of permitted program binaries, processes, scripts and benign network activity.

In academic literature, Rizvi et al. present this as a distributed approach for a firewall system in automotive networks [77]. Their system is focused on allowing only authorised packets to reach an internal device using a Hybrid Security System (HSS) that uses many individual firewalls located in front of each module and at each electronic unit.

The inspiration for our study comes from a paper Klement et al. [39], where the authors propose a fine-grained filtering mechanism on the CAN-BUS in a vehicle, exploiting a man-in-the-middle vulnerability incorporating a firewall inside the dongle (see Figure 6). According to the analysis results from [39], the proposed approach allows a fine-grained and extensible filtering approach for all protocols within the OBD.

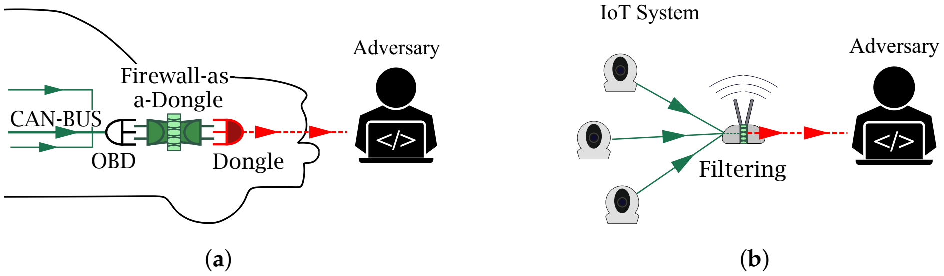

Figure 6. Data leakage from (**a**) the car via OBD [39,40] and from (**b**) the IoT system.

4. Goal: Filtering for Increased Privacy

In this research, we investigate the impact of using filtering as a technical method for protecting personal data. While notwithstanding its use within other jurisdictions or with other definitions of privacy, we suggest taking the European data protection law codified in EU Directive 95/46/EC [78] to define *personal data* to 'mean any information relating to an identified or identifiable natural person (*data subject*)' [78]. Europe's General Data Protection Regulation (Directive 95/46/EC [78]) requires one —among other things—to 'minimise the amount of collected data' [79].

Alongside the possible application of encryption to ensure the confidentiality of such personal data, we see the filtering (i.e., by a firewall) as a means to minimise data leakage. The intent is to block the transmission of data as early as possible, i.e., closer to the points where data are created, thereby allowing the user to enforce the data minimisation principle.

While we use the technical concept of packet-filtering (see Section 2.4), one could even go further and use content re-writing rules to perform data anonymisation on the actual data within a packet. Not allowing some information to flow might be tolerated for some applications, and hence, 'the data collected [. . .] should be strictly necessary for the specific purpose previously determined by the data controller (the 'data minimisation' principle)' [79].

Following the principle of privacy by design [80,81] should lead to fewer data becoming collected or data leakage to be minimal [82]. However, in reality, users are unable, incapable [48] or unwilling to change the devices themselves for those that offer increased privacy; filtering can be adopted. Filtering demands a trusted device that allows the firewalling functions to be run inside the critical network communication. In the Smart Home, this is usually the router or wireless access points, minimising the amount of transferred unsecured data, e.g., data sent from sensors or the system to a back-end. This effort being spent to prevent accidental disclosure is sometimes called *data loss prevention* or *data leakage prevention* (DLP) [83]. Such a data leakage in the car and Smart Home domain is shown in Figure 6. It clearly shows that for privacy, we need to prohibit information from flowing in the opposite direction, as in Figure 4.

Network and security architectures are developed so that the devices achieve security mostly without user interaction in the sense of 'security by default'. However, we would like the user to become involved in crucial security decisions but not bothered with everyday requests, especially if they might impact security, as this can lead to *warning fatigue* and also security fatigue in general [84**?**]. We assume that guiding a user to effectively filter certain traffic or encapsulation of devices can be employed to combat warning fatigue in IT security.

We want to provide users with the availability to have a fine-grained choice if they wish, as they are the only ones who can make privacy decisions in everyday situations.

This would also require including tools for users to dynamically adjust or change their choices [86].

However, we believe that the network and security architectures must provide a high level of security even without requiring constant and skilful user interaction, as many users may not have enough knowledge to properly manage their own security settings, and a system that can provide security without relying on user input can be more effective at preventing cyberattacks.

5. Scenario Descriptions

In this section, the following four scenarios, which depict the previously discussed diverse communication patterns in the Smart Home domain are presented:

α: Communication with the Internet;
β: Local Network Communication;
γ: Guest Access;
δ: Indirect Control.

Each of these is first motivated, described, and illustrated in a Data Flow Diagram (DFD) [28,31]. We also indicate the trust boundary of the *local network*, which is the network segment located behind the home gateway that performs Network Address Translation (NAT). Thus, adversaries from outside are not able to communicate with devices in the local network unless it is made possible by means of port-forwarding rules, either in the router configuration or through other techniques like uPnP (Universal Plug and Play [87]).

5.1. Scenario α: Essential and Unessential Communication with the Internet

In general, we can distinguish two types of devices based on how tied to the Internet they are: (a) devices that can and (b) devices that cannot be used without an Internet connection [16]. Digging deeper into devices that rely on Internet connectivity, we can see that they communicate with both mandatory (e.g., updates) and not mandatory services (e.g., crush reports).

This scenario therefore focuses on this particular aspect, where a *Smart Home Device* might communicate with multiple servers on the Internet, some of which contribute to its main functionality and some of which do not; the latter communication would therefore be dispensable.

While in reality both categories might comprise multiple servers, the scenario abstracts from this by replacing them with two distinct servers, each of them representing one of the two groups. The first server, representing the servers to which functionality-related messages are sent, is called *Essential Server*. In contrast, the *Unessential Server* represents the group of servers that only handle communication which is not contributing to the user experience. Notice that a device might communicate with the same server for multiple purposes, possibly including both functional and non-functional communication. The *Smart Home Device* is in this scenario the sole appliance inside the user's local network in order to put the focus on the distribution of outgoing connections. This scenario is depicted in Figure 7. Notice that the information flow between the *Smart Home Device* and the *Unessential Server* is unidirectional. This reflects the assumption that the communication is merely used to collect information (e.g., crash reports, logs, usage statistics) from the device.

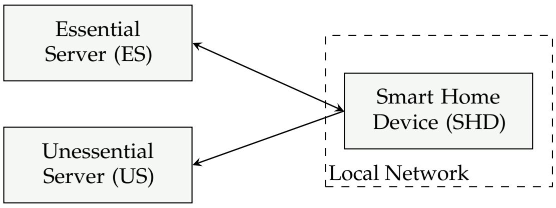

Figure 7. Data Flow Diagram of Scenario α (communication with the Internet).

5.2. Scenario β: Local Network Communication

This scenario focuses on communication within the local network, where several *Smart Home Devices* communicate with the *Owner Control Device*. *Server X* and *Server Y* represent the limited sets of connections a *Smart Home Device* uses during normal (i.e., not compromised) operation. The *Owner Control Device*, belonging to the general-purpose computing device category, is in contrast to the *Smart Home Devices* simply connected to the Internet as a whole. Notice that *Server X* and *Server Y* are also part of the Internet and might therefore be accessed by the *Owner Control Device*.

Additionally, we assume that the *Owner Control Device* is trusted for two reasons. First, it seems irrational that a homeowner would intentionally attack their own devices. Second, if the *Owner Control Device* is compromised, and considering its unlimited access to all devices in the network, this would make any filtering or other mitigation method obsolete.

This scenario is depicted in Figure 8, where the Internet is shown in grey because it is not part of the further threat modelling, as it covers a too broad space to be modelled meaningfully. Also, the branch of *Smart Home Device Y* and *Server Y* is not part of the modelling and is therefore greyed out because it is assumed that there cannot be any new threats for those devices and corresponding data flows compared to the equally arranged branch of *Smart Home Device X* and *Server X*.

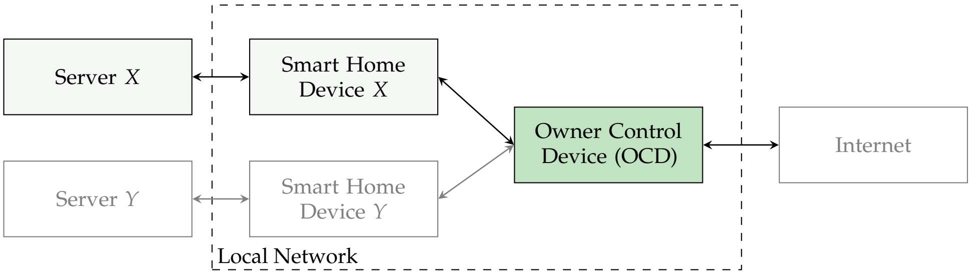

Figure 8. Data Flow Diagram of Scenario β (Local Network Communication).

5.3. Scenario γ: Guest Access

Typically, when a guest visits the homeowner, they request access to WiFi. It is possible that the guest has access only to the Internet without connection to the network including *Smart Home Devices*. However, assume that the homeowner wants to give a guest access not only to the Internet but also to one specific *Smart Home Device*, e.g., a light switch. In this case, the guest's device is placed in the same network, which includes not only the target device but also others. Additionally, we assume that the homeowner does not want a guest to interact with any devices or even know of their existence apart from the target one. Some *Smart Home Devices* might offer an authorisation mechanism that restricts who is connecting to them. Others might simply consider a device in the local network as trustworthy.

This scenario is depicted in Figure 9. The *Guest Control Device* controls the *Accessible Smart Home Device*. The *Private Smart Home Device* represents the devices that are not to be controlled by the guest. Similar to the *Owner Control Device*, the *Guest Control Device* is also connected to the entire Internet, assuming that this is most likely a general-purpose computing device. e.g., a Smartphone. Additionally, we assume, that the devices do not need to communicate with each other and that no threats can originate from the *Owner Control Device*.

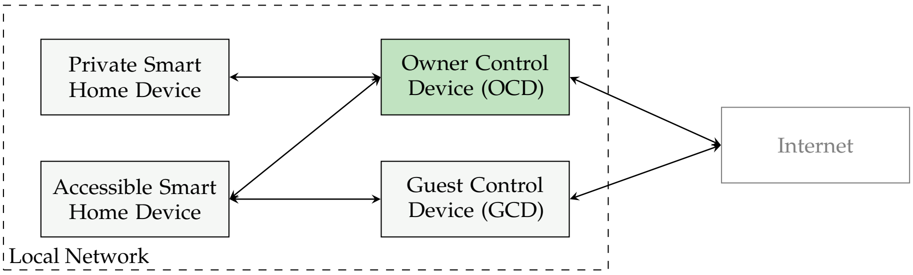
Private Smart Home Device
Owner Control Device (OCD)
Internet
Accessible Smart Home Device
Guest Control Device (GCD)
Local Network

Figure 9. Data Flow Diagram of Scenario γ (guest access).

5.4. Scenario δ: Indirect Control

Some *Smart Home Devices* can be controlled remotely via an external server, which acts as a relay between the controlling device of the user and the *Smart Home Devices*. It allows the user to not be within the local network to perform control, but at the same time, it restricts the router of the firewall from determining the responsible item for the communication to the device in the local network.

The scenario is depicted in Figure 10 and actually combines two closely related scenarios. The first one covers the indirect control of a *Smart Home Device* by a device in the same network. The second scenario covers the remote control of the *Smart Home Device* by a *Remote Control Device* which is outside the local network. In this scenario, the *Remote Control Device* is the owner Smartphones/tablet/PC, etc. For simplicity, it is assumed that the *Smart Home Device* is limited to indirect control and therefore communicates only with a predefined *Control Server*, which relays the commands coming from other devices. There is however no assumption to what extent this restriction is enforced by the device (i.e., if the communication is authenticated).

The *Remote Control Device* and its communication with the *Control Server* are shown in grey because they are not part of the threat modelling later on, as both are outside the local network and therefore out of the scope of the techniques considered throughout this paper.

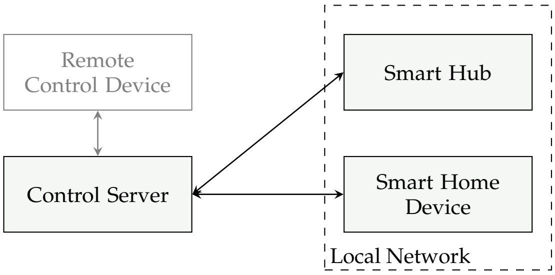

Figure 10. Data Flow Diagram of Scenario δ (indirect control).

6. STRIDE and LINDDUN Analysis of Threats in the Smart Home Domain

This section focuses on our analysis of the impact of filtering on the four scenarios (see Section 5) and their data flows from the Smart Home domain. We identify possible attacks and their impacts using the modified versions of STRIDE per element for security and LINDDUN per element for privacy-related threats. Threat analysis was performed by manually analysing each component involved in the scenario, possible attack vectors, and their impact. We believe that since the scenarios used are generalised architectures typical for Smart Home, the results can be applied to create filtering rules regardless of how they are produced, manually by the user, by copying, or by AI (see Section 2.5).

In order to clearly present the results of our threat analysis, we provide comprehensive tables for each scenario that include possible threats and related impacts and reflect the influence of filtering. Each threat (spoofing, tampering, likability, etc.) is presented in the form of a table. Several tables are combined with each other with respect to the analysis technique (STRIDE or LINDDUN). In general, there are the following columns (see Figure 11):

Threat Target is a device or communication line that is the subject of a threat (e.g., Smart Home Device, Smart Hub, CS → SHD, etc.).
Threat Impact includes the possible influence of the threat on the ***Threat Target*** (e.g., a device can receive a malicious firmware update).
Attack Vectors contain one or more possible attack vectors leafing to the associated *Threat Impact* (e.g., an attacker being able to send the malicious firmware by impersonating the update server using stolen credentials).

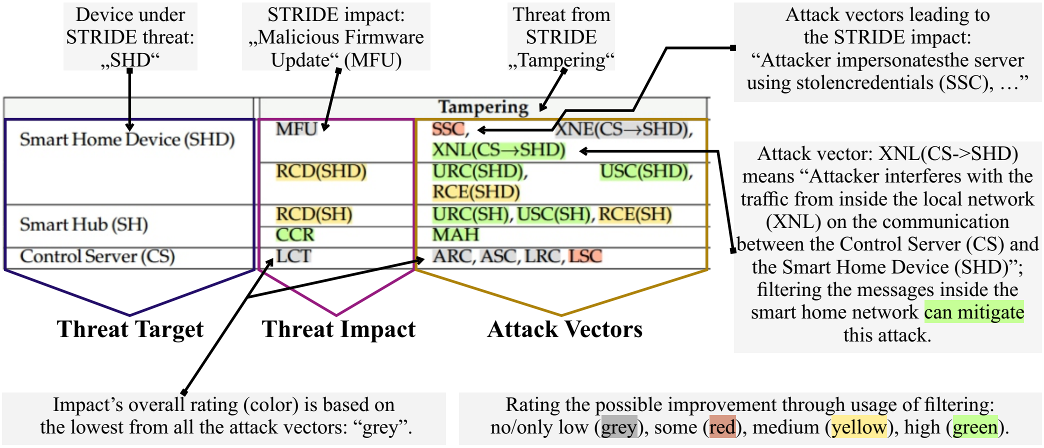
Device under STRIDE threat: „SHD“
STRIDE impact: „Malicious Firmware Update“ (MFU)
Threat from STRIDE „Tampering“
Attack vectors leading to the STRIDE impact: “Attacker impersonatesthe server using stolencredentials (SSC), …”
Tampering
Smart Home Device (SHD)
Smart Hub (SH)
Control Server (CS)
MFU
RCD(SHD)
RCD(SH)
CCR
LCT
SSC, XNE(CS→SHD), XNL(CS→SHD)
URC(SHD), USC(SHD), RCE(SHD)
URC(SH), USC(SH), RCE(SH)
MAH
ARC, ASC, LRC, LSC
Attack vector: XNL(CS->SHD) means “Attacker interferes with the traffic from inside the local network (XNL) on the communication between the Control Server (CS) and the Smart Home Device (SHD)”; filtering the messages inside the smart home network can mitigate this attack.
Threat Target
Threat Impact
Attack Vectors
Impact’s overall rating (color) is based on the lowest from all the attack vectors: “grey”.
Rating the possible improvement through usage of filtering: no/only low (grey), some (red), medium (yellow), high (green).

Figure 11. Explanation of the STRIDE and LINDDUN threat tables used to show the impact and attack vectors and colour coding to indicate the potential positive impact of filtering in this scenario.

In order to visualise the impact of filtering, a traffic-light-coloured rating with four scales is used:

No or only low (grey) influence means that the application of filtering either has absolutely no effect on the fulfilment of the attack or has such a negligible effect that the attack is still successful and has the same impact as without filtering. On the other end of the scale, a *High* (green) influence of filtering is determined when, if it is applied, the attack cannot be successfully completed or the impact of a successful attack is negligible.

Based on these ratings, the influence on STRIDE and LINDDUN impacts in the tables is aggregated using the minimum level that was assessed for any of the vectors leading to it (see the impacts in the second column as shown in Figure 11).

For compactness, the impacts and attack vectors are symbolised by a sequence of three characters, which can be seen in Appendix A in Tables A1 and A2 for STRIDE and Tables A3 and A4 for LINDDUN.

Note that we skipped the *Information Disclosure* threat in LINDDUN modelling to avoid unnecessary redundancy, as that threat is already included in STRIDE. Additionally, the *Non-Repudiation* threat of LINDDUN is not present while the *Repudiation* threat of STRIDE is. In our evaluation, we assume that non-repudiation (non-repudiation is the goal to which repudiation is the threat in STRIDE terminology) has a higher priority from a security point of view than plausible deniability. (Plausible deniability is the goal to which non-repudiation is the threat in the LINDDUN terminology. This assumption can be challenged. For example, from a privacy perspective, a resident may want to hide their actions from others; e.g., if a teenager is leaving their parent's house late at night but wants to hide this from his parents. Deciding whose goals are more important, the teenager's goal to be able to leave the house undetected or the parent's goal to know about their children's activities, is, however, out of the scope of our analysis.) Thus, we emphasise in our analysis the security threat—which would endanger the ability to prove which entity triggered what action—over the privacy aspect.

6.1. Scenario α (Communication with the Internet)

In this scenario, essential and unessential communication of a *Smart Home Device* with the Internet is discussed (see Figure 7). The result is depicted in Tables 3 and 4.

6.1.1. Scenario α: STRIDE

Spoofing:

As a result of exploiting the threat of spoofing against *SHD*, wrong actions can be taken by the device (*DWA*), like opening the front door even if not intended by a legitimate user. This is feasible if the device is susceptible to spoofing of the server's identity, e.g., by sending it messages with a spoofed IP address (*XNE(ES→SHD)* (notice that the actual vector is *INE(ES→SHD)*, which is, however, condensed in the corresponding XNE vector with other network interference vectors for compactness, which is justified by their close relation)). However, if the device can check the authenticity of messages, stolen valid credentials of the server itself, like a leaked private key (*SSC*), pose a threat that is considered unlikely. If there are no integrity protections applied to the messages, an attacker in the right position (e.g., a national agency or the Internet Service Provider (ISP)) might also tamper with the traffic from the essential server to the Smart Home Device (*XNE(ES→SHD)*) outside the local network to achieve the same goal.

Considering *ES* as the threat target, an attacker can try to impersonate *SHD* by using stolen or brute-forced credentials (*DSC*, *DRC*). If *ES* obtains the wrong information/data, it can lead to an incorrect response action, e.g., an unnecessary call to the fire service (*SWA*).

Tampering:

Similar to spoofing, the same attack vectors can lead to severe tampering threats, like a malicious firmware update (*MFU*) that is sent to the device and applied with the expectation that it originates from the trusted server. In that case, an attacker might gain full control over the device.

Another threat to the integrity arises if an attacker can run commands on the device to either turn it into a bot or compromise its functionality, even from outside the local network (*RCD*). This might be possible by means of logging into the device with stolen (*USC*), default, or otherwise simple-to-guess passwords (*URC*). Additionally, a device might offer a web interface that might suffer from a Remote Code Execution vulnerability (*RCE*) that allows an attacker to issue commands on the device.

Repudiation:

Two impacts are listed for the Repudiation category: the server might deny either the sending of a message (*SDS*), which leads to an unwanted action (e.g., opening the door without the user's intent) or the receipt of a message (*SRD*), which should have triggered an action (e.g., calling the fire department). Both of these threats can arise if there is no sufficient logging of messages deployed on the device itself or in the network (*DIL*), or if the messages themselves are not authenticated using, e.g., digital signatures (*SMU*).

Information Disclosure:

If the *SHD* itself collects data, the extraction of potentially private information might be possible (*EPI*) by using the same attack vectors (i.e., *USC*, *URC*, or *RCE*) as before. However, not only might the device itself leak private information, but if data are stored on a server (*ES*, *US*), that server is also a potential target for threatening confidentiality. The severity of an attack depends highly on the nature and amount of data collected by the server, which might be less severe logs concerning the device's functionality or very personal data or videos (*SCD(ES)*, *SCD(US)*). An attacker might, e.g., be able to read data from a company's database, if they are accessible from the Internet and not protected sufficiently, or worse, not all (*DBO(ES)*, *DBO(US)*). For example, the latter case has been reported [88] for a toy manufacturer, which sold teddy bears that allowed children to send messages to their parents and vice versa. While the corresponding database did not contain the recordings itself, it had the links to the records, hashed user passwords, and other information such as e-mail addresses.

Targeting communication channels, a similar impact (*SCD(SHD→ES)*, *SCD(SHD→US)*), even if limited to the time of exposure and probably requiring special attackers (e.g., a national agency or the ISP), might be achieved when the traffic from the device to any of the servers can be eavesdropped (*ENE(SHD→ES)*, *ENE(SHD→US)*) from outside the local network.

DoS:

Not only may a device carry out wrong actions, as described before, but also the absence of such (*DNA*) can be a threat, e.g., if the device does not open the door for a legitimate user or does not raise an alarm in case of an emergency. This might be achieved by performing any kind of Denial of Service attack (*DOS(SHD)*) or a WLAN de-authentication attack (*WDA*) on the device itself. Dropping the traffic from the *ES* to the *SHD* from outside the local network (*DNE(ES→SHD)*) might also be an option if an attacker is able to perform such actions (e.g., a national agency or the ISP). For a device to lose part of its functionality in addition to those attack vectors, a DoS attack on the *ES* (*DOS(ES)*) or dropping the communication to it (*DNE(SHD→ES)*) might also be a threat, e.g., in case of an assistant that can no longer evaluate the commands of the user if the server responsible for that task is not available.

Elevation of Privilege:

A rather unusual threat is the one presented for this category, namely gaining control over the *SHD* by gaining control over the server first (*IDC*), where the attack vector is the usage of stolen or guessed admin credentials on the *ES* (*ASC(ES)*, *ARC(ES)*). Similarly, obtaining access to the *US* (*ASC(US)*, *ARC(US)*) might not lead to the ability to control the device directly but to the ability to gain knowledge (e.g., authentication tokens) that can be used to perform further attacks (*NRO*).

It is worth noticing that one of the few cases where there is a difference between the essential and the unessential servers is the threat of no action (*SNA*) or a wrong action by the server (*SWA*), which can be explained by considering that the unessential server is not associated with the functionality of the device, and hence its behaviour has no influence on it.

6.1.2. Scenario α: LINDDUN

Linkability:

Successful information disclosure attacks (see Section 6.1.1) enable an attacker to exploit threats in the linkability category, leading to the creation of a profile of the user's activity. That way, an attacker might not only see when a user goes to bed once but be able to predict future behavior.

Data might be linked on the device (*LDD*) or on a server (*LDS(ES)*, *LDS(US)*) directly (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied). This is of course possible for the operator of the server but may also be possible for an attacker in case of an information disclosure threat (*IDS(SHD)*, *IDS(ES)*, *IDS(US)*).

Table 3. STRIDE table for Scenario α (internet connectivity) showing the influence of filtering. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Spoofing		**Tampering**	
Smart Home Device (SHD)	DWA	SSC, XNE(ES→SHD)	MFU	SSC, XNE(ES→SHD)
			RCD	URC, USC, RCE
Essential Server (ES)	SWA	DRC, DSC, XNE(SHD→ES)		
	Repudiation		**Information Disclosure**	
Smart Home Device (SHD)	SDS	DIL, SMU	EPI	URC, USC, RCE
	SDR	DIL, SMU		
Essential Server (ES)			SCD(ES)	DBO(ES)
Unessential Server (US)			SCD(US)	DBO(US)
SHD → ES			SCD(SHD→ES)	ENE(SHD→ES)
SHD → US			SCD(SHD→US)	ENE(SHD→US)
	Denial of Service		**Elevation of Priviledge**	
Smart Home Device (SHD)	DNA	DOS(SHD), DNE(ES→SHD), DNE(SHD→ES)—for DLF, WDA		
	DLF			
Essential Server (ES)	SNA	DOS(ES), DNE(SHD→ES), WDA	IDC	ARC(ES), ASC(ES)
Unessential Server (US)			NRO	ARC(US), ASC(US)

Threat Impacts

DLF The Smart Home Device loses its functionality (totally or partially)

DNA The Smart Home Device obtains wrong or no information and therefore does not carry out an action (e.g., unlocking the door or raising an alarm)

DWA The Smart Home Device obtains wrong information and in turn makes wrong actions (e.g., unlocking the door or raising an alarm)

EPI Attacker extracts private information from the device

IDC Attacker obtains control over the Smart Home Device indirectly by controlling an entity that already has control over it

MFU A malicious firmware update is sent to the Smart Home Device

NRO Not relevant on its own but may lead to other threats

SCD Depending on the data collected by the server, e.g., not critical (logs) or critical, once revealing private information or video material

RCD Attacker runs commands on the device to damage it or turn it into a bot

SNA The server obtains wrong or no information and therefore does not carry out an action (e.g., buying something or calling the fire department)

SDR Server denies the receipt of a message (which would have triggered actions like e.g., buying something or calling the fire department)

SDS Server denies the sending of a message (which might have triggered actions like unlocking the door or raising an alarm)

SWA The server obtains wrong information and in turn makes wrong actions (e.g., buying something or calling the fire department)

Attack Vectors

ARC Attacker logs into the server with randomly guessed admin credentials

ASC Attacker logs into the server with stolen admin credentials

DBO Database stores information open (i.e., accessible from outside and not protected or only weakly protected)

DIL There is no or no sufficient logging of messages or acknowledgements from others on the device or in the network

DNE	Attacker outside the local network drops traffic using an MitM position (MME)
DOS	Any form of Network Denial of Service attack (e.g., DDoS using SYN Flooding)
DRC	Attacker tries random credentials or tokens to impersonate the Smart Home Device
DSC	Attacker impersonates the Smart Home Device with stolen credentials
ENE	Attacker outside the local network eavesdrops on traffic using an MitM position (MME)
RCE	An interface (e.g., web interface) of the device has a Remote Code Execution vulnerability
SMU	Messages from the server are not authenticated (e.g., with Digital Signatures, Message authentication codes are not sufficient)
SSC	Attacker impersonates the server using stolen credentials (e.g., leaked private key)
URC	Attacker tries random or default credentials to log into one of the device's user accounts (potentially root) (even behind NAT, this might be possible if the device installed an open port using uPnP)
USC	Attacker logs into one of the device's user accounts (potentially root) using stolen credentials (even behind NAT, this might be possible if the device installed an open port using uPnP)
WDA	Attacker performs a WLAN deauthentication attack on the device
XNE	Attacker interferes with the traffic from outside the local network (DNE, INE, RNE, MNE)

Closely related is the ability of an attacker to link messages coming from the *SHD* together (*LMD(SHD→ES)*, *LMD(SHD→US)*). This can be achieved by the same means as are necessary for detection, namely collecting metadata (*CME*, *CMM*) in order to identify what device sent or received a message (which is trivial in this scenario, as only one device is present in the network).

Identifiability:

Identifying the user of a device itself (*IUD*) might be possible if an identifier like an e-mail address or physical properties of a person is stored together with other data under the assumption of a successful exploit information disclosure threat *IDS(SHD)*, similar to identifying whose data are stored on the server (*IUS(ES)*, *IUS(US)*). Additionally, identifying a device's owner can be achieved if the messages sent from or to the device are not or only weakly encrypted and contain identifiers as the ones mentioned before (*MRD(SHD)*), which, basically, also arises from the threat of information disclosure (see Section 6.1.1).

Detectability:

Detecting the presence of devices in the home is referred to as an inventory attack (*INA*), which can be achieved by collecting metadata about the device communication either from messages leaving the home network (*CME*) or by watching the potentially encrypted WLAN packets (*CMM*). With the same kind of attacks, an attacker might also infer a user's activity (*IUA*) (e.g., if there is a peak in the communication of a sleep monitor when the user goes to bed [18] or when the door is opened).

Unawareness:

Closely related is the threat of content unawareness, as the user might not know which data are collected by a device or sent outside the home network (*UCD*) because they are not informed in an appropriate way about it (*DCN*).

Noncompliance:

Threats to policy or consent non-compliance arise if the server collects more information than the user agreed on or might be reasonable (*MDS(ES)*, *MDS((US)*) or if such information becomes available to a third party (*MDT(ES)*, *MDT(US)*). It can happen in several cases: a manufacturer built the device in such a way that they send data to either the manufacturer's server (*BDS(ES)*, *BDS(US)*)or a third party (*BDT(ES)*, *BDT(US)*) directly, and if one of the servers shares the data it received (*BST(ES)*, *BST(US)*).

Additionally, the device itself might collect more data than a user intended (*MDD*) if it was built in such a way (*BDC*) that can be seen as the basis for sending the data in the way described before, or as an amplifier for other threats, e.g., when thinking of the threat of extracting private information (*EPI*) as described before in the STRIDE section (see Section 6.1.1).

In contrast to STRIDE threat modelling, no differences were found between essential and non-essential servers, meaning that in terms of privacy, they pose the same number of threats while being functionally irrelevant.

Table 4. LINDDUN table for Scenario α (internet connectivity) showing the influence of filtering. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Linkability		**Identifiability**	
Smart Home Device (SHD)	LDD	IDS(SHD)	IUD	IDS(SHD), MRD(SHD)
Essential Server (ES)	LDS(ES)	IDS(ES)	IUS(ES)	IDS(ES)
Unessential Server (US)	LDS(US)	IDS(US)	IUS(US)	IDS(US)
SHD → ES	LMD(SHD→ES)	CME, CMM		
SHD → US	LMD(SHD→US)	CME, CMM		

	Detectability		**Unawareness**		**Noncompliance**	
Smart Home Device (SHD)	INA	CME, CMM	UCD	DCN	MDD	BDC
	IUA	CME, CMM				
Essential Server (ES)					MDS(ES)	BDS(ES)
					MDT(ES)	BDT(ES), BST(ES)
Unessential Server (US)					MDS(US)	BDS(US)
					MDT(US)	BDT(US), BST(US)

Threat Impacts

INA Inventory attack (i.e., the attacker gets to know that there is a device of the type of the Smart Home Device at home)

IUA Infer a user's activity (e.g., when a user goes to bed)

IUD Identify the user of the device (e.g., when an identifier like an email address or physical properties of a person is stored on the device)

IUS Identify who's data is stored on the server (e.g., when an identifier like an email address or physical properties of a person is stored on the server together with usage data)

LDD Link data on the Smart Home Device (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)

LDS Link data on the server (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)

LMD Link messages of the device (e.g., at different points in time to create a profile of when the user is asleep)

MDD The device itself collects more information than the user agreed on or is reasonable

MDS The server collects more information than the user agreed on or is reasonable

MDT There is more information passed to third parties than the user agreed on or is reasonable

UCD User is unaware of the data collected or sent by the Smart Home Device

Attack Vectors

BDS Manufacturer built the device in a way such that it sends data it should not send to the server

BDT Manufacturer built the device in a way such that it sends data it should not send to a third party

BST Manufacturer built the server in a way such that it sends data it should not send to a third party

CME An attacker outside the local network collects metadata of the network traffic by observing network traffic leaving the home using an MitM position (MME) and compares them to patterns from known devices

CMM Attacker outside the local network collects metadata of the network traffic by observing encrypted WLAN traffic and compares them to patterns from known devices

DCN The user has not been informed about the data collected

IDS Information disclosure of the corresponding entry in the STRIDE modelling

MRD The messages sent by or sent to the device are readable (IDS(d → x), IDS(x → d)) and contain identifiers such as an email address or physical properties of a person

6.2. Scenario β (Local Network Communication)

While this scenario shares many threats with the previous one, it also contains new ones due to the introduction of a new device type—*Owner Control Device (OCD)* (see Figure 8). Hence, in this section, we discuss only new aspects that are different from the ones presented in Section 6.1.1. The results are presented in Tables 5 and 6.

We assume that *OCD* is trusted, but its communication might require added security, as well as the need to consider attacks against *OCD* originating from inside the local network, e.g., in the case of the presence of an already infected *SHD*. The latter might be the result of an attack prior to the installation of a suitable mitigation technique or the acquisition of an already corrupted device [56,89].

6.2.1. Scenario β: STRIDE

Spoofing:

A new threat related to *OCD* is related to the possibility that *OCD* may have a display and thus can show wrong status information about the *SHD* it is meant to control (*WSI*) to the user. This can arise if the status updates sent from the Smart Home Device are dropped or tampered with (*XNL(SHD→OCD)*) or if the devices are disconnected from each other by a WLAN de-authentication attack on any of them (*WDA(SHD)*, *WDA(OCD)*).

Tampering:

As an *OCD*-related threat, we can identify the possibility of tampering with communication between *OCD* and the Internet (*XNE(OCD→I)*, *XNL(OCD→I)* and *XNE(I→OCD)*, *XNL(I→OCD)*). The associated impact, however, depends on many factors (*DEP*), such as the usage of the control device or the security level of the websites it connects to (e.g., no Internet access, no file syncing with remote client/server, session fixation/hijacking when browsing, etc.).

Information Disclosure:

Potentially, an attacker can log messages (clear/encrypted) that are transferred between *SHD* and *OCD* (*LLM(OCD→SHD)*, *LLM(SHD→OCD)*). This can be achieved by performing man-in-the-middle (MitM) attacks (*MML(OCD→SHD)*, *MML(SHD→OCD)*).

Possible MitM attack types, also for *MME*, are taken from the recent comprehensive survey provided by Conti et al. in [90]. Additionally, the threat of a rogue access point that tries to imitate the genuine one, also referred to as an Evil Twin, has been considered. (We tested the susceptibility of a TuxWang smart socket regarding this threat. First, it was assigned to a WPA2-protected access point realised by a Raspberry Pi 4B running `hostapd` (c.f. https://www.raspberrypi.com/documentation/computers/configuration.html (last accessed: 19 January 2025)). After changing the `hostapd` configuration to offer no protection at all, both the laptop running `Linux Mint 19.2` as well as the Smartphones running Android 9 refused to connect to it, but the outlet connected to the network successfully, which was verified by controlling it remotely over Smartphones. Additionally, disconnecting the device with WLAN-deauthentication, which might be necessary to make the device reconnect, was proven to be possible in the WPA2 setup using `bettercap v2.24` (https://www.bettercap.org/ (last accessed: 19 January 2025)).

Local and external MitM attacks allow an attacker to intercept traffic on the local and the Internet interfaces. Moreover, an attacker might obtain access to credentials or tokens (*OCR*) sent in plain text over the network by eavesdropping on the messages between the two devices in the local network (*ENL(OCD→SHD)*, *ENL(SHD→OCD)*).

Also interesting is the threat of disclosed information about a user's Internet activity in the form of clear data (*CDD(OCD→I)*, *CDD(I→OCD)*) or metadata (*CDM(OCD→I)*, *CDM(I→OCD)*), likely revealing the websites a user visits in the form of IP addresses or

DNS queries or responses. This can be achieved by eavesdropping in or outside the local network on the corresponding data flows (*ENE(OCD→I)*, *ENL(OCD→I)* and *ENE(I→OCD)*, *ENL(I→OCD)*).

DoS:

Not only can *SHD* lose functionality by attacking its network traffic, but so can *OCD* (*CLF*). If *OCD* is a Smartphone, the Internet-related functionality may be most of what it offers. Disconnecting the device from the Internet or the local network could be achieved by performing a DoS or WLAN de-authentication attack directly on the device (*DOS(OCD)*, *WDA(OCD)*) or by dropping the network traffic as an MitM device (*DNE(OCD→I)*, *DNL(OCD→I)*, *DNE(I→OCD)*, *DNL(I→OCD)*). It should be noted that, while a Smartphone might use the mobile network as an alternative, making it more robust to such attacks, this is unlikely to be the case for most traditional computers (note that a user might provide a hotspot with their phone to circumvent this problem).

In addition, many attack vector collections have been extended by the possibility of interfering with the network traffic not only from outside but also from inside the network (*XNL*).

Table 5. STRIDE table for Scenario β (Local Network Connectivity) showing the influence of filtering. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Spoofing	
Owner Control Device (OCD)	WSI	XNL(SHD→OCD), WDA(SHD), WDA(OCD)
Smart Home Device (SHD) *X*	DWA	SSC, XNL(OCD→SHD), XNE(S→SHD), XNL(S→SHD)
Server (S) *X*	SWA	DRC, DSC, XNE(SHD→S), XNL(SHD→S)

	Tampering		**Repudiation**	
Owner Control Device (OCD)	RCD(OCD)	URC(OCD)		
Smart Home Device (SHD) *X*	MFU	SSC, XNE(S→SHD), XNL(S→SHD)	SDS	DIL, SMU
	RCD(SHD)	URC(SHD), USC(SHD), RCE	SDR	DIL, SMU
OCD → Internet	DEP(OCD→I)	XNE(OCD→I), XNL(OCD→I)		
Internet → OCD	DEP(I→OCD)	XNE(I→OCD), XNL(I→OCD)		

	Information Disclosure		**Elevation of Priviledge**	
Owner Control Device (OCD)	EPI(OCD)	URC(OCD)		
Smart Home Device (SHD) *X*	EPI(SHD)	URC(SHD), USC(SHD), RCE		
	OCR	ENL(OCD→SHD), ENL(SHD→OCD)		
Server (S) *X*	SCD(S)	DBO	IDC	ARC, ASC
OCD → Internet	CDD(OCD→I)	ENE(OCD→I), ENL(OCD→I)		
	CDM(OCD→I)	ENE(OCD→I), ENL(OCD→I)		
Internet → OCD	CDD(I→OCD)	ENE(I→OCD), ENL(I→OCD)		
	CDM(I→OCD)	ENE(I→OCD), ENL(I→OCD)		
OCD → SHD	LLM(OCD→SHD)	MML(OCD→SHD)		
SHD → OCD	DCD	ENL(SHD→OCD)		
	LLM(SHD→OCD)	MML(SHD→OCD)		
SHD → S	SCD(SHD→S)	ENE(SHD→S), ENL(SHD→S)		

	Denial of Service	
Owner Control Device (OCD)	CLF	DOS(OCD), WDA(OCD), DNE(OCD→I), DNL(OCD→I), DNE(I→OCD), DNL(I→OCD)
Smart Home Device (SHD) *X*	DNA	DOS(SHD), WDA(SHD), WDA(OCD), DNL(OCD→SHD), DNE(S→SHD), DNL(S→SHD)
	DLF	DOS(SHD), DOS(S), WDA(SHD), DNE(SHD→S), DNL(SHD→S), DNE(S→SHD), DNL(S→SHD)
Server (S) *X*	SNA	DOS(S), WDA(SHD), DNE(SHD→S), DNL(SHD→S)

Threat Impacts

CDD Disclosure of the data sent or received by the control device

CDM Disclosure of the metadata of messages sent or received by the control device (this may e.g., reveal visited web pages)

CLF The control device loses Internet-related functionality (which may be most of the functionality e.g., for a Smartphone)

DCD Depends on the data sent by the Smart Home Device, which might be less severe data like logs concerning the device's functionality or very sensitive data revealing private information or video material

DEP Depends highly on the usage of the control device or the security level of the websites it connects to (e.g., no Internet access, no file syncing with remote client/server, session fixation/hijacking when browsing, etc.)

DLF The Smart Home Device loses its functionality (totally or partially)

DNA The Smart Home Device obtains wrong or no information and therefore does not carry out an action (e.g., unlocking the door or raising an alarm)

DWA The Smart Home Device obtains wrong information and in turn makes wrong actions (e.g., unlocking the door or raising an alarm)

EPI Attacker extracts private information from the device

IDC Attacker obtains control over the Smart Home Device indirectly by controlling an entity that already has control over it

LLM Attacker can log (potentially encrypted) messages in the local network (e.g., to replay them later on)

MFU A malicious firmware update is sent to the Smart Home Device

OCR Attacker obtains credentials or tokens used for authentication of the user

RCD Attacker runs commands on the device to damage it or turn it into a bot

SCD Depending on the data collected by the server, which might be less severe data like logs concerning the device's functionality or very sensitive data revealing private information or video material

SDR Server denies receipt of a message (which would have triggered actions like e.g., buying something or calling the fire department)

SDS Server denies the sending of a message (which might have triggered actions like unlocking the door or raising an alarm)

SNA The server obtains wrong or no information and therefore does not carry out an action (e.g., buying something or calling the fire department)

SWA The server obtains wrong information and in turn makes wrong actions (e.g., buying something or calling the fire department)

WSI The device has wrong status information about a Smart Home Device it controls

Attack Vectors

ARC Attacker logs into the server with randomly guessed admin credentials

ASC Attacker logs into the server with stolen admin credentials

DBO Database stores information open (i.e., accessible from outside and not protected or only weakly protected)

DIL There is no or no sufficient logging of messages or acknowledgements from others on the device or in the network

DNE Attacker outside the local network drops traffic using an MitM position (MME)

DNL Attacker inside the local network drops traffic using an MitM position (MML)

DOS Any form of Network Denial of Service attack (e.g., DDoS using SYN-Flooding)

DRC Attacker tries random credentials or tokens to impersonate the Smart Home Device

DSC Attacker impersonates the Smart Home Device with stolen credentials

ENE Attacker outside the local network eavesdrops on traffic using an MitM position (MME)

ENL Attacker inside the local network eavesdrops on traffic using an MitM position (MML)

MML Attacker performs a man-in-the-middle attack inside the local network (ARP spoofing, DHCP spoofing, Evil Twin)

RCE An interface (e.g., web interface) of the device has a Remote Code Execution vulnerability

SSC Attacker impersonates the server using stolen credentials (e.g., leaked private key)

URC Attacker tries random or default credentials to log into one of the device's user accounts (potentially root) (even behind NAT, this might be possible if the device installed an open port using uPnP)

USC Attacker logs into one of the device's user accounts (potentially root) using stolen credentials (even behind NAT this might be possible if the device installed an open port using uPnP)

WDA Attacker performs a WLAN deauthentication attack on the device

XNE Attacker interferes with the traffic from outside the local network (DNE, INE, RNE, MNE)

XNL Attacker interferes with the traffic from inside the local network (DNL, INL, RNL, MNL)

6.2.2. Scenario β: LINDDUN

Identifiability:

Identifying the user of the *OCD* might, apart from the information disclosure (*IDS(OCD)*) and the readable messages containing identifiers (*MRD(OCD)*), which have already been covered for the *SHD* in the previous scenario, also be possible if the device has an identifying name (e.g., Tim's iPhone) (*IDN(OCD)*) that it includes in messages it sends (e.g., in DHCP requests).

Detectability:

In addition to the attack vectors already described for S_α, inventory attacks (*INA(OCD)*, *INA(SHD)*) can also be performed by scanning the home network from inside (*NSL*), e.g., by using arp or pings manually, or tools that are similar to 'nmap'. Another way of detecting devices can be to monitor broadcast messages in the local network (*CBL*), which is helpful if those messages contain information that allows the identification of the device type.

Unawareness:

Not only can the *SHD* send data that the user is unaware of but the software on the *OCD*, which is used to control the *SHD*, can also send data to the manufacturer or a third party without the user's awareness (*UCC*). This is again the case if the user was not informed about the data which are sent or collected (*DCN*).

In addition, many attack vector collections have been extended by the possibility to discover network traffic not only from outside but also from inside the network (*CML*).

Table 6. LINDDUN table for Scenario β (Local Network Connectivity) showing the influence of filtering. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Linkability		**Identifiability**	
Owner Control Device (OCD)	LDC	IDS(OCD)	IUD(OCD)	IDS(OCD), MRD(OCD), IDN(OCD)
Smart Home Device (SHD) *X*	LDD	IDS(SHD)	IUD(SHD)	IDS(SHD), MRD(SHD)
Server (S) *X*	LDS	IDS(S)	IUS	IDS(S)
OCD → Internet	LMD(OCD→I)	IDS(OCD→I)	IMC(OCD→I)	IUI(OCD→I)
Internet → OCD	LMD(I→OCD)	IDS(I→OCD)	IMC(I→OCD)	IUI(I→OCD)
OCD → SHD	LMD(OCD→SHD)	CMM, CML	IMC(OCD→SHD)	IUI(OCD→SHD), IUI(SHD→OCD)
SHD → OCD	LMD(SHD→OCD)	CMM, CML	IMC(SHD→OCD)	IUI(SHD→OCD), IUI(OCD→SHD)
SHD → S	LMD(SHD→S)	CME, CMM, CML		

	Detectability		**Unawareness**		**Noncompliance**	
Owner Control Device (OCD)	INA	CMM, CML, CBL, NSL	UCC	DCN	MDD	BCC
	IUA	CME, CMM, CML				
Smart Home Device (SHD) *X*	INA	CME, CMM, CML, CBL, NSL	UCD	DCN	MDD	BDC
	IUA	CME, CMM, CML				
Server (S) *X*					MDS	BDS, BCS
					MDT	BDT, BST, BCT

Threat Impacts

IMC Identify the user based on messages sent to or by the control device.

INA Inventory attack (i.e., the attacker gets to know that there is a device of the type of the Smart Home Device at home)

IUA Infer a user's activity (e.g., when a user goes to bed)

IUD Identify the user of the device (e.g., when an identifier like an email address or physical properties of a person is stored on the device)

LDC	Link data on the control device (because of data probably stored in user-specific folders on a multi-user PC or only one user on single-user devices like Smartphones, this will in most cases be trivial)
LDD	Link data on the Smart Home Device (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)
LDS	Link data on the server (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)
LMD	Link messages of the device (e.g., at different points in time to create a profile of when the user is asleep)
MDD	The device itself collects more information than the user agreed on or is reasonable
MDS	The server collects more information than the user agreed on or is reasonable
MDT	There is more information passed to third parties than the user agreed on or is reasonable
UCC	User is unaware of the data sent by the control/companion app of the Smart Home Device
UCD	User is unaware of the data collected or sent by the Smart Home Device
	Attack Vectors
BCC	Manufacturer built the control/companion app in a way such that it collects data it should not
BCS	Manufacturer built the control/companion app in a way such that it sends data it should not send to the server
BCT	Manufacturer built the control/companion app in a way such that it sends data it should not send to a third party
BDC	Manufacturer built the device in a way such that it collects data it should not
BDS	Manufacturer built the device in a way such that it sends data it should not send to the server
BDT	Manufacturer built the device in a way such that it sends data it should not send to a third party
BST	Manufacturer built the server in a way such that it sends data it should not send to a third party
CBL	Attacker collects local broadcast messages which contain information that identifies a device
CME	Attacker outside the local network collects metadata of the network traffic by observing network traffic leaving the home using an MitM position (MME) and compares them to patterns from known devices
CML	Attacker inside the local network collects metadata of the network traffic using an MitM position (MML) and compares them to patterns from known devices
CMM	Attacker outside the local network collects metadata of the network traffic by observing encrypted WLAN traffic and compares them to patterns from known devices
DCN	The user has not been informed about the data collected
IDN	The device has an identifying device name (e.g., Tim's iPhone)
IDS	Information disclosure of the corresponding entry in the STRIDE modelling
IUI	The user is identified by linking the source or destination address with the corresponding device's user
MRD	The messages sent by or sent to the device are readable (IDS(d → x), IDS(x → d)) and contain identifiers such as an email address or physical properties of a person
NSL	Attacker scans the internal network (e.g., using arp or pings, perhaps by using a tool like nmap) for available hosts or open ports and services offered by a host.

6.3. Scenario γ (Guest Access)

This scenario is dedicated to describing threats that can be present in the system to which an owner's guest can have access (see Figure 9). The results are shown in Tables 7 and 8. Because of the high number of devices and data flows, also the number of entries in the analysis tables is also relatively high compared to the other scenarios. Conversely, the number of new threats is, however, fairly low. This can be explained by the fact that the newly introduced Guest Control Device (*GCD*) is closely related to the Owner Control Device (*OCD*), which has already been included in the previous scenario. Even more, the earlier consideration of attacks inside the local network in Scenario S_β (see Section 6.2) leaves only a little space for new threats.

It shows that there may not be a need to deal with the different types of attackers on the network in different ways, but that the same or at least similar measures may be effective against both corrupted *SHDs* and misbehaving guests at the same time.

6.3.1. Scenario γ: STRIDE

Repudiation:

One of the few novel threats can be identified within the repudiation category. This category of threat is concerned with the situation in which a guest denies having interacted with either of the two Smart Home Devices, namely *GDS(PSHD)* and *GDS(ASHD)*. In the second case, only the denial of certain messages may be of relevance. However, in the former case, which pertains to the *PSHD*, any interaction with this device by a guest is unintended and therefore constitutes a potential threat. Insufficient logging may also contribute to the emergence of such threats ((*DIL(PSHD)*, *DIL(ASHD)*).

Elevation of Privilege:

That a guest accesses the *PSHD* (*GAP*) is considered an elevation of privilege threat as they are not intended to interact with it but only with the *ASHD*. This threat can arise if there is no access control insufficient access control on the device (*MAC(PSHD)*). An additional threat in this category is that the guest obtains remote control (while the ability to control a device remotely is only considered explicitly in the later Scenario S_δ, in that scenario, there will be no guest devices anymore, so the threat fits best into this one) or permanent control over the device when no or only local or temporal control is intended by the homeowner (*RAC(PSHD)*, *RAC(ASHD)*). This is possible in the case of missing authorisation checks on the server relaying the remote communication (*MAS(PSHD)*, *MAS(ASHD)*).

The other changes in analysis tables are mostly related to the different configurations in this scenario (e.g., no server, additional control device) but do not contain new threats.

Table 7. STRIDE table for Scenario γ (guest access) showing the influence of filtering. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Spoofing	
Owner Control Device (OCD)	WSI(OCD)	XNL(PSHD→OCD), WDA(PSHD), WDA(OCD), XNL(ASHD→OCD), WDA(ASHD)
Guest Control Device (GCD)	WSI(GCD)	XNL(ASHD→GCD), WDA(ASHD), WDA(GCD)
Private SHD	DWA(PSHD)	SSC, XNL(OCD→PSHD)
Accessible SHD	DWA(ASHD)	SSC, XNL(OCD→ASHD), XNL(GCD→ASHD)

	Tampering		**Repudiation**	
Owner Control Device (OCD)	RCD(OCD)	URC(OCD)		
Guest Control Device (GCD)	RCD(GCD)	URC(GCD)		
Private SHD	RCD(PSHD)	URC(PSHD), USC(PSHD), RCE(PSHD)	SDS(PSHD)	DIL(PSHD), SMU
			SDR(PSHD)	DIL(PSHD), SMU
			GDS(PSHD)	DIL(PSHD)
Accessible SHD	RCD(ASHD)	URC(ASHD), USC(ASHD), RCE(ASHD)	SDS(ASHD)	DIL(ASHD), SMU
			SDR(ASHD)	DIL(ASHD), SMU
			GDS(ASHD)	DIL(ASHD)
OCD → Internet	DEP(OCD→I)	XNE(OCD→I), XNL(OCD→I)		
Internet → OCD	DEP(I→OCD)	XNE(I→OCD), XNL(I→OCD)		
GCD → Internet	DEP(GCD→I)	XNE(GCD→I), XNL(GCD→I)		
Internet → GCD	DEP(I→GCD)	XNE(I→GCD), XNL(I→GCD)		

Table 7. *Cont.*

	Information Disclosure	
Owner Control Device (OCD)	EPI(OCD)	URC(OCD)
Guest Control Device (GCD)	EPI(GCD)	URC(GCD)
Private SHD	EPI(PSHD)	URC(PSHD), USC(PSHD), RCE(PSHD)
	OCR(PSHD)	ENL(OCD→PSHD), ENL(PSHD→OCD)
Accessible SHD	EPI(ASHD)	URC(ASHD), USC(ASHD), RCE(ASHD)
	OCR(ASHD)	ENL(OCD→ASHD), ENL(ASHD→OCD), ENL(GCD→ASHD), ENL(ASHD→GCD)
OCD → Internet	CDD(OCD→I)	ENE(OCD→I), ENL(OCD→I)
	CDM(OCD→I)	ENE(OCD→I), ENL(OCD→I)
Internet → OCD	CDD(I→OCD)	ENE(I→OCD), ENL(I→OCD)
	CDM(I→OCD)	ENE(I→OCD), ENL(I→OCD)
GCD → Internet	CDD(GCD→I)	ENE(GCD→I), ENL(GCD→I)
	CDM(GCD→I)	ENE(GCD→I), ENL(GCD→I)
Internet → GCD	CDD(I→GCD)	ENE(I→GCD), ENL(I→GCD)
	CDM(I→GCD)	ENE(I→GCD), ENL(I→GCD)
OCD → PSHD	LLM(OCD→PSHD)	MML(OCD→PSHD)
PSHD → OCD	DCD(PSHD→OCD)	ENL(PSHD→OCD)
	LLM(PSHD→OCD)	MML(PSHD→OCD)
OCD → ASHD	LLM(OCD→ASHD)	MML(OCD→ASHD)
ASHD → OCD	DCD(ASHD→OCD)	ENL(ASHD→OCD)
	LLM(ASHD→OCD)	MML(ASHD→OCD)
GCD → ASHD	LLM(GCD→ASHD)	MML(GCD→ASHD)
ASHD → GCD	DCD(ASHD→GCD)	ENL(ASHD→GCD)
	LLM(ASHD→GCD)	MML(ASHD→GCD)

	Denial of Service		**Elevation of Privilege**	
Owner Control Device (OCD)	CLF(OCD)	DOS(OCD), WDA(OCD), DNE(OCD→I), DNL(OCD→I), DNE(I→OCD), DNL(I→OCD)		
Guest Control Device (GCD)	CLF(GCD)	DOS(GCD), WDA(GCD), DNE(GCD→I), DNL(GCD→I), DNE(I→GCD), DNL(I→GCD)		
Private SHD	DNA(PSHD)	DOS(PSHD), WDA(PSHD), WDA(OCD), DNL(OCD→PSHD)	RAC(PSHD)	MAS(PSHD)
	DLF	DOS(PSHD)	GAP	MAC(PSHD)
Accessible SHD	DNA(ASHD)	DOS(ASHD), WDA(ASHD), DNL(OCD→ASHD), WDA(OCD), WDA(GCD), DNL(GCD→ASHD)	RAC(ASHD)	MAS(ASHD)
	DLF	DOS(ASHD)		

Threat Impacts

CDD Disclosure of the data sent or received by the control device

CDM Disclosure of the metadata of messages sent or received by the control device (this may e.g., reveal visited web pages)

CLF The control device loses Internet-related functionality (which may be most of the functionality e.g., for a Smartphone)

DCD Depends on the data sent by the Smart Home Device, which might be less severe data like logs concerning the device's functionality or very sensitive data revealing private information or video material

DEP Depends highly on the usage of the control device or the security level of the websites it connects to (e.g., no Internet access, no file syncing with remote client/server, session fixation/hijacking when browsing, etc.)

DLF The Smart Home Device loses its functionality (totally or partially)

DNA The Smart Home Device obtains wrong or no information and therefore does not carry out an action (e.g., unlocking the door or raising an alarm)

DWA The Smart Home Device obtains wrong information and in turn makes wrong actions (e.g., unlocking the door or raising an alarm)

EPI	Attacker extracts private information from the device
GAP	Guest accesses the Private Smart Home Device
GDS	Guest denies the sending of a message (which might have triggered actions like unlocking the door or raising an alarm)
LLM	Attacker can log (potentially encrypted) messages in the local network (e.g., to replay them later on)
OCR	Attacker obtains credentials or tokens used for authentication of the user
RAC	Guest obtains remote access to the Smart Home Device when only local or no access was intended
RCD	Attacker runs commands on the device to damage it or turn it into a bot
SDR	Server denies receipt of a message (which would have triggered actions like e.g., buying something or calling the fire department)
SDS	Server denies the sending of a message (which might have triggered actions like unlocking the door or raising an alarm)
WSI	The device has wrong status information about a Smart Home Device it controls
	Attack Vectors
DIL	There is no or no sufficient logging of messages or acknowledgements from others on the device or in the network
DNE	Attacker outside the local network drops traffic using an MitM position (MME)
DNL	Attacker inside the local network drops traffic using an MitM position (MML)
DOS	Any form of Network Denial of Service attack (e.g., DDoS using SYN-Flooding)
ENE	Attacker outside the local network eavesdrops on traffic using an MitM position (MME)
ENL	Attacker inside the local network eavesdrops on traffic using an MitM position (MML)
MAC	Missing access control on the Smart Home Device
MAS	Missing authorisation checks on the server (e.g., when registering for remote access in an app)
MML	Attacker performs a man-in-the-middle attack inside the local network (ARP spoofing, DHCP spoofing, Evil Twin)
RCE	An interface (e.g., web interface) of the device has a Remote Code Execution vulnerability
SMU	Messages from the server are not authenticated (e.g., with Digital Signatures, Message authentication codes are not sufficient)
SSC	Attacker impersonates the server using stolen credentials (e.g., leaked private key)
URC	Attacker tries random or default credentials to log into one of the device's user accounts (potentially root) (even behind NAT this might be possible if the device installed an open port using uPnP)
USC	Attacker logs into one of the device's user accounts (potentially root) using stolen credentials (even behind NAT this might be possible if the device installed an open port using uPnP)
WDA	Attacker performs a WLAN deauthentication attack on the device
XNE	Attacker interferes with the traffic from outside the local network (DNE, INE, RNE, MNE)
XNL	Attacker interferes with the traffic from inside the local network (DNL, INL, RNL, MNL)

6.3.2. Scenario γ: LINDDUN

Only two new privacy threats have been discovered for this scenario.

Unawareness:

The first new threat is related to the unawareness category and lies in the fact that the guest is unaware of the data or metadata collected or logged by the homeowner (*GCH*). This is the case, if the homeowner does not communicate the way traffic is logged or observed in the network (e.g., on the router), if at all (*DCN*).

Noncompliance:

The second new threat is closely related to the previous one. In this case, there are more data collected by the homeowner than the user agreed to or is reasonable (*HMD*). This threat can arise if the homeowners themselves are unaware of the data which are collected and potentially transmitted to a remote server for analysis (*HUC*) or if the homeowner intentionally monitors the activity of devices in the home network (*HSD*). This latter scenario is not considered a significant threat but is included to demonstrate that even the homeowner has the potential to pose a threat to other actors within the network.

The remaining modifications are mainly associated with the distinct configuration in this scenario (e.g., the absence of a server, the inclusion of an additional control device), but they do not introduce any novel threats.

Table 8. LINDDUN table for Scenario γ (Guest access) showing the influence of filtering. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Linkability		**Identifiability**	
Owner Control Device (OCD)	LDC(OCD)	IDS(OCD)	IUD(OCD)	IDS(OCD), MRD(OCD), IDN(OCD)
Guest Control Device (GCD)	LDC(GCD)	IDS(GCD)	IUD(GCD)	IDS(GCD), MRD(GCD), IDN(GCD)
Private Smart Home Device (PSHD)	LDD(PSHD)	IDS(PSHD)	IUD(PSHD)	IDS(PSHD), MRD(PSHD)
Accessible Smart Home Device (ASHD)	LDD(ASHD)	IDS(ASHD)	IUD(ASHD)	IDS(ASHD), MRD(ASHD)
OCD → Internet	LMD(OCD→I)	IDS(OCD→I)	IMC(OCD→I)	IUI(OCD→I)
Internet → OCD	LMD(I→OCD)	IDS(I→OCD)	IMC(I→OCD)	IUI(I→OCD)
GCD → Internet	LMD(GCD→I)	IDS(GCD→I)	IMC(GCD→I)	IUI(GCD→I)
Internet → GCD	LMD(I→GCD)	IDS(I→GCD)	IMC(I→GCD)	IUI(I→GCD)
OCD → PSHD	LMD(OCD→PSHD)	CMM, CML	IMC(OCD→PSHD)	IUI(OCD→PSHD)
PSHD → OCD	LMD(PSHD→OCD)	CMM, CML	IMC(PSHD→OCD)	IUI(PSHD→OCD)
OCD → ASHD	LMD(OCD→ASHD)	CMM, CML	IMC(OCD→ASHD)	IUI(OCD→ASHD)
ASHD → OCD	LMD(ASHD→OCD)	CMM, CML	IMC(ASHD→OCD)	IUI(ASHD→OCD)
GCD → ASHD	LMD(GCD→ASHD)	CMM, CML	IMC(GCD→ASHD)	IUI(GCD→ASHD)
ASHD → GCD	LMD(ASHD→GCD)	CMM, CML	IMC(ASHD→GCD)	IUI(ASHD→GCD)

	Detectability		**Unawareness**		**Noncompliance**	
Owner Control Device (OCD)	INA	CMM, CML, CBL, NSL	UCC	DCN	MDD	BCC(OCD)
	IUA	CME, CMM, CML				
Guest Control Device (GCD)	INA	CMM, CML, CBL, NSL	UCC	DCN	MDD	BCC(GCD)
	IUA	CME, CMM, CML	GCH	DCN	HMD	HUC, HSD
Private SHD	INA	CMM, CML, CBL, NSL	UCD	DCN	MDD	BDC(PSHD)
	IUA	CMM, CML				
Accessible SHD	INA	CMM, CML, CBL, NSL	UCD	DCN	MDD	BDC(ASHD)
	IUA	CMM, CML				
OCD → Internet					MDS	BCS(OCD)
					MDT	BCT(OCD)
GCD → Internet					MDS	BCS(GCD)
					MDT	BCT(GCD)

Threat Impacts

GCH Guest is unaware of the data or metadata collected/logged by the homeowner

INA Inventory attack (i.e., the attacker gets to know that there is a device of the type of the Smart Home Device at home)

IUA Infer a user's activity (e.g., when a user goes to bed)

IUD Identify the user of the device (e.g., when an identifier like an email address or physical properties of a person is stored on the device)

LDC Link data on the control device (because of data probably stored in user-specific folders on a multi-user PC or only one user on single-user devices like Smartphones, this will in most cases be trivial)

LDD Link data on the Smart Home Device (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)

LDS Link data on the server (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)

LMD Link messages of the device (e.g., at different points in time to create a profile of when the user is asleep)

UCC User is unaware of the data sent by the control/companion app of the Smart Home Device

UCD User is unaware of the data collected or sent by the Smart Home Device

Attack Vectors

BCC Manufacturer built the control/companion app in a way such that it collects data it should not

BCS Manufacturer built the control/companion app in a way such that it sends data it should not send to the server

BCT Manufacturer built the control/companion app in a way such that it sends data it should not send to a third party

BDC Manufacturer built the device in a way such that it collects data it should not

CME Attacker outside the local network collects metadata of the network traffic by observing network traffic leaving the home using an MitM position (MME) and compares them to patterns from known devices

CML Attacker inside the local network collects metadata of the network traffic using an MitM position (MML) and compares them to patterns from known devices

CMM Attacker outside the local network collects metadata of the network traffic by observing encrypted WLAN traffic and compares them to patterns from known devices

DCN The user has not been informed about the data collected

IDN The device has an identifying device name (e.g., Tim's iPhone)

IDS Information disclosure of the corresponding entry in the STRIDE modelling

IUI The user is identified by linking the source or destination address with the corresponding device's user

MRD The messages sent by or sent to the device are readable (IDS(d → x), IDS(x → d)) and contain identifiers such as an email address or physical properties of a person

NSL Attacker scans the internal network (e.g., using arp or pings, perhaps by using a tool like nmap) for available hosts or open ports and services offered by a host.

6.4. Scenario δ (Indirect Control)

In this scenario, indirect control over *Smart Home Device (SHD)* is discussed (see Figure 10). The results are shown in Tables 9 and 10. *Smart Hub (SH)* is introduced as a device that does not control the *SHD* directly but via a server. However, it borrows threats from both *SHDs* and *Control Devices*. Therefore, some but not many completely new threats arise from this constellation. Furthermore, the indirect control mechanism itself introduces additional attack vectors, resulting in a cumulative set of relevant threats, some of which are outlined below.

6.4.1. Scenario δ: STRIDE

Spoofing:

The ability to control the device remotely naturally also brings new threats with it, like the ability of an attacker to do the same as what a legitimate user is permitted (*CDR*) by logging into their account using stolen (*LSC*) or guessed credentials (*LRC*).

Tampering:

One of the new tampering threats related to the newly introduced *SH* is that an attacker could change the control routines on *SH* (*CCR*) if no suitable authorisation checks are in place (*MAH*). Moreover, an attacker can delete the account of a legitimate user (*LCT*) by logging in (*LRC*, *LSC*) using user's or admin credentials (*ARC*, *ASC*) the same way as it is already described in the analysis of Scenario S_α (see Section 6.1). It should be noted that the server may implement measures to prevent this event from occurring. One potential method is to require confirmation of the action through access to a secret link sent to the user's email account.

The rather unusual threat of controlling the *SHD* indirectly over a server, as described for Scenario S_α, is getting more realistic when relayed by the *SH* instead (*IDC(SH)*). This goal might be achieved by exploiting a Remote Code Execution vulnerability on the *SH* (*RCE(SH)*) or logging into it using stolen, default, or otherwise simple-to-guess passwords (*URC(SH)*, *USC(SH)*).

Information Disclosure:

An attacker can also obtain status information about the *SHD* from the *SH* (*SID*) by logging into it (*URC(SH)*, *USC(SH)*) or by logging into the user account on the server (*LRC*, *LSC*) as described before using stolen or guessed credentials. There might also be credentials stored on the hub used for authentication of the commands which can be disclosed (*OCR*) by again logging into it or eavesdropping on the traffic from the hub to the server either from inside the network or outside (*ENE(SH→CS)*, *ENL(SH→CS)*).

Table 9. STRIDE table for Scenario δ (indirect control) showing the influence of filtering. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Spoofing	
Smart Home Device (SHD)	DWA	SSC, XNE(CS→SHD), XNL(CS→SHD), XNE(SH→CS), XNL(SH→CS)
Smart Hub (SH)	WSI	XNE(CS→SH), XNL(CS→SH), WDA(SH)
Control Server (CS)	SWA	DRC, DSC, XNE(SHD→CS), XNL(SHD→CS), XNE(SH→CS), XNL(SH→CS)
	CDR	LRC, LSC

	Tampering		**Repudiation**	
Smart Home Device (SHD)	MFU	SSC, XNE(CS→SHD), XNL(CS→SHD)	SDS(SHD)	DIL(SHD), SMU
	RCD(SHD)	URC(SHD), USC(SHD), RCE(SHD)	SDR(SHD)	
Smart Hub (SH)	RCD(SH)	URC(SH), USC(SH), RCE(SH)	SDR(SH)	DIL(SH)
	CCR	MAH		
Control Server (CS)	LCT	ARC, ASC, LRC, LSC		

	Information Disclosure		**Elevation of Priviledge**	
Smart Home Device (SHD)	EPI(SHD)	URC(SHD), USC(SHD), RCE(SHD)		
	SID	URC(SH), USC(SH), LRC, LSC		
Smart Hub (SH)	OCR	URC(SH), USC(SH), ENE(SH→CS), ENL(SH→CS)	IDC(SH)	RCE(SH), URC(SH), USC(SH)
Control Server (CS)	SCD(CS)	DBO(CS)	IDC(CS)	ARC, ASC
SHD → CS	SCD(SHD→CS)	ENE(SHD→CS), ENL(SHD→CS)		
SH → CS	SCD(SH→CS)	ENE(SH→CS), ENL(SH→CS)		

	Denial of Service	
Smart Home Device (SHD)	DNA	DOS(SHD), DOS(CS), DOS(SH), WDA(SHD), WDA(SH), DNE(CS→SHD), DNL(CS→SHD), DNE(SH→CS), DNL(SH→CS)
	DLF	DOS(SHD), DOS(CS), WDA(SHD), DNE(SHD→CS), DNL(SHD→CS), DNE(CS→SHD), DNL(CS→SHD)
Control Server (CS)	SNA	DOS(CS), WDA(SHD), DNE(SHD→CS), DNL(SHD→CS), WDA(SH), DNE(SH→CS), DNL(SH→CS)
	LCD	DOS(CS)

Threat Impacts

CCR Changing the control routines

CDR Attacker might remotely control the Smart Home Device

DLF The Smart Home Device loses its functionality (totally or partially)

DNA The Smart Home Device obtains wrong or no information and therefore does not carry out an action (e.g., unlocking the door or raising an alarm)

DWA The Smart Home Device obtains wrong information and in turn makes wrong actions (e.g., unlocking the door or raising an alarm)

EPI Attacker extracts private information from the device

IDC Attacker obtains control over the Smart Home Device indirectly by controlling an entity that already has control over it

LCD A legitimate user (e.g., Smart Hub or Remote Control Device) loses control over the Smart Home Device because the control server is unavailable

LCT A legitimate user (e.g., Smart Hub or Remote Control Device) loses control over the Smart Home Device because the attacker deleted their account information on the Control server

MFU A malicious firmware update is sent to the Smart Home Device

OCR Attacker obtains credentials or tokens used for the authentication of the user

RCD Attacker runs commands on the device to damage it or turn it into a bot

SCD Depending on the data collected by the server, which might be less severe data like logs concerning the device's functionality or very sensitive data revealing private information or video material

SDR Server denies receipt of a message (which would have triggered actions like e.g., buying something or calling the fire department)

SDS Server denies the sending of a message (which might have triggered actions like unlocking the door or raising an alarm)

SID Attacker obtains status information about the Smart Home Device from another entity

SNA The server obtains wrong or no information and therefore does not carry out an action (e.g., buying something or calling the fire department)

SWA The server obtains wrong information and in turn makes wrong actions (e.g., buying something or calling the fire department)

WSI The device has wrong status information about a Smart Home Device it controls

Attack Vectors

ARC Attacker logs into the server with randomly guessed admin credentials

ASC Attacker logs into the server with stolen admin credentials

DBO Database stores information open (i.e., accessible from outside and not protected or only weakly protected)

DIL There is no or no sufficient logging of messages or acknowledgements from others on the device or in the network

DNE Attacker outside the local network drops traffic using an MitM position (MME)

DNL Attacker inside the local network drops traffic using an MitM position (MML)

DOS Any form of Network Denial of Service attack (e.g., DDoS using SYN-Flooding)

DRC Attacker tries random credentials or tokens to impersonate the Smart Home Device

DSC Attacker impersonates the Smart Home Device with stolen credentials

ENE Attacker outside the local network eavesdrops on traffic using an MitM position (MME)

ENL Attacker inside the local network eavesdrops on traffic using an MitM position (MML)

LRC Attacker tries random credentials to log into a legitimate user account on the server

LSC Attacker logs into a legitimate user account on the server with stolen credentials

MAH Missing authorisation checks on the hub (e.g., when adding rules)

RCE An interface (e.g., web interface) of the device has a Remote Code Execution vulnerability

SMU Messages from the server are not authenticated (e.g., with Digital Signatures, Message authentication codes are not sufficient)

SSC Attacker impersonates the server using stolen credentials (e.g., leaked private key)

URC Attacker tries random or default credentials to log into one of the device's user accounts (potentially root) (even behind NAT this might be possible if the device installed an open port using uPnP)

USC Attacker logs into one of the device's user accounts (potentially root) using stolen credentials (even behind NAT this might be possible if the device installed an open port using uPnP)

WDA Attacker performs a WLAN deauthentication attack on the device

XNE Attacker interferes with the traffic from outside the local network (DNE, INE, RNE, MNE)

XNL Attacker interferes with the traffic from inside the local network (DNL, INL, RNL, MNL)

6.4.2. Scenario δ: LINDDUN

Detectability:

One of the few new privacy-related threats identified for this scenario is the additional attack vector of detecting the *SHD* (*INA(SHD)*) through the communication of the *SH* with the *CS* (*ICM*). The possibility of this occurring may depend on the architecture used by the *SH*. This could be determined by whether the *SHD* directly communicates with the manufacturer's server or is communicating with a single server that then talks to the *SHD* manufacturer's server. The latter seems to be the case for the Amazon Alexa Smart Home Skill API [91].

Noncompliance:

Noncompliance refers to the case that the Smart Hub might collect or send more information than the user agreed on or is reasonable to the manufacturer of the hub (e.g., about the usage of devices under control) (*MDH*) because it was built that way (*BDC(SH)*), which is especially a problem if the user is unaware of that (*UCH*) as they were not informed about that (*DCN*).

Table 10. LINDDUN table for Scenario δ (indirect control) showing the influence of filtering. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Linkability		**Identifiability**	
Smart Home Device (SHD)	LDD	IDS(SHD)	IUD(SHD)	IDS(SHD), MRD(SHD)
Smart Hub (SH)	LDH	IDS(SH)	IUD(SH)	IDS(SH), MRD(SH)
Control Server (CS)	LDS	IDS(CS)	IUS	IDS(CS)
SHD → CS	LMD(SHD→CS)	CME, CMM, CML		
SH → CS	LMD(SH→CS)	CME, CMM, CML		

	Detectability		**Unawareness**		**Noncompliance**	
Smart Home Device (SHD)	INA	CME, CMM, CML, CBL, NSL, ICM	UCD	DCN	MDD	BDC(SHD)
	IUA	CME, CMM, CML				
Smart Hub (SH)	INA	CME, CMM, CML, CBL, NSL	UCH	DCN	MDH	BDC(SH)
	IUA	CME, CMM, CML				
Control Server (CS)					MDS	BDS(SHD), BDS(SH)
					MDT	BDT(SHD), BDT(SH), BST

Threat Impacts

INA Inventory attack (i.e., the attacker gets to know that there is a device of the type of the Smart Home Device at home)

IUA Infer a user's activity (e.g., when a user goes to bed)

IUD Identify the user of the device (e.g., when an identifier like an email address or physical properties of a person is stored on the device)

LDD Link data on the Smart Home Device (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)

LDH Link data on the Smart Hub (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)

LDS Link data on the server (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)

MDD The device itself collects more information than the user agreed on or is reasonable

MDH The Smart Hub collects or sends more information than the user agreed on or is reasonable (e.g., about the usage of devices under control to the manufacturer of the hub)

MDS The server collects more information than the user agreed on or is reasonable

MDT There is more information passed to third parties than the user agreed on or is reasonable

Attack Vectors

BDC Manufacturer built the device in a way such that it collects data it should not

BDS Manufacturer built the device in a way such that it sends data it should not send to the server

BDT Manufacturer built the device in a way such that it sends data it should not send to a third party

BST Manufacturer built the server in a way such that it sends data it should not send to a third party

CBL Attacker collects local broadcast messages which contain information that identifies a device

CME Attacker outside the local network collects metadata of the network traffic by observing network traffic leaving the home using an MitM position (MME) and compares them to patterns from known devices

CML Attacker inside the local network collects metadata of the network traffic using an MitM position (MML) and compares them to patterns from known devices

CMM Attacker outside the local network collects metadata of the network traffic by observing encrypted WLAN traffic and compares them to patterns from known devices

IDS Information disclosure of the corresponding entry in the STRIDE modelling

MRD The messages sent by or sent to the device are readable (IDS(d → x), IDS(x → d)) and contain identifiers such as an email address or physical properties of a person

NSL Attacker scans the internal network (e.g., using arp or pings, perhaps by using a tool like nmap) for available hosts or open ports and services offered by a host.

7. Discussion of the Filtering Impact on Security and Privacy

In this section, we discuss the influence of a filtering approach on the scenarios under consideration and how it reflects on STRIDE and LINDDUN analysis results. We consider that the technology used for filtering would be at least as fine-grained in their rules as the technique referred to in this paper as *Bark* [26], presented by Hong et al.; a short overview about *Bark* and its potential impact on performance and data quality is given in Section 2.4 and Section 2.6, respectively.

7.1. Influence of Filtering on STRIDE Attack Vectors

At first, the influence of network filtering according to *Bark* on the STRIDE attack vectors, as given in the Appendix A (see Table A1), is discussed. Because the effectiveness of filtering-based approaches highly depends on the rules which are in place to filter traffic, scenario-specific circumstances need to be considered. Therefore, scenario-specific adjustments of the influence ratings are discussed after the vector-specific discussions.

ARC, ASC, DRC, DSC, LRC, LSC:

During the analysis, it was identified that measures that restrict traffic only within the local network, or entering/leaving the home router, are not able to mitigate any login attempts on a remote server.

There might be an influence on the process of stealing the credentials in the first place, e.g., if stolen using a local MitM attack (not applicable to admin passwords); but on the other hand, vectors like a leaked password database are invariant. (*DSC* and *LSC* only appear on servers other than the *Unessential server*; therefore, even if for the latter an influence is assumed, this does not apply to the servers for which those two attack vectors are relevant.) Therefore, again, the influence is rated as *no or only low* for the attack vectors *ARC*, *ASC*, *DRC* and *LRC* but as *some* for *DSC* and *LSC*.

DBO:

Similarly, the presence of vulnerabilities on the server side, like a database that is disclosed to the public or only weakly protected, is also out of scope for filtering approaches installed on the home router. Therefore, an influence rating of *no or only low* is assigned for this vector (*DBO*).

DIL:

Filtering has no positive influence on logging, leading to a rating of *no or only low* (*DIL*).

DNE, DNL, ENE, ENL, MNE, MNL:

Filtering rules cannot prevent an MitM from intercepting, eavesdropping on, or modifying traffic, but it can prevent an attacker from getting into a position suitable to perform MitM attack types. Then, we can conclude that the level of filtering influence is the same as for the MitM vector (*MME* or *MML*) they are based on. Therefore, the influence on *DNE*, *ENE* and *MNE* is rated as *no or only low*, but a *medium* rating is assigned for the *DNL*, *ENL* and *MNL* vectors.

Note, following a defensive in-depth strategy, an additional end-to-end encryption with proper authentication of the endpoint [92] prevents a man-in-the-middle attacker from being able to directly read the traffic. However, even when communication is encrypted, *traffic analysis* [4,5] can violate privacy. This attack can reveal information without the knowledge of the communication's contents. Of course, proper authenticated end-to-end encrypted channels are still too often neglected, so attackers can use MitM instead of having to perform more costly and potentially unsuccessful traffic analysis [93]. Still, even with encryption, such attacks may allow attackers to undermine the security of the person [8–10]. However, we conclude that filtering is helpful by preventing an attacker from getting into a position that would allow them to carry out the MitM as discussed above because this would at the same time prevent an attacker from being in the position to observe the traffic for a successful traffic analysis attack.

DOS:

The *Bark* filtering approach can prevent DoS attack traffic if it is not whitelisted. However, there is a possibility that wired clients, which are not considered in the approach, can bypass the filtering mechanisms by spoofing their MAC addresses. There is also the possibility of indirectly targeting a victim via a device that already has network access and is additionally accessible to the attacker. Lyu et al. have shown that all eight devices they tested can be used to reflect TCP SYN attacks, and some of them even support high gain reflection [94]. Therefore, the impact is only rated as *some* (*DOS*). The exception is the case of a DoS attack that targets not one of the local devices but a remote server, in which case the impact is rated as *no or low* (*DOS*).

INE, RNE:

The insertion of traffic directed to a destination outside the local network, such as a control server, cannot be prevented by the home router.

The opposite direction is also unlikely to have a great impact, assuming an attacker spoofs a whitelisted address. The effect is then limited to reducing the number of endpoints that can be spoofed for a device. Overall, therefore, a rating of *no or low* impact is assigned for both directions (*INE*, *RNE*).

INL, RNL:

As *Bark* does not appear to enforce a correct mapping between MAC and IP addresses, a host can in principle send packets with spoofed IP addresses. However, similar to DoS attacks, the *Bark* filtering approach can prevent the insertion of traffic, benign or malicious, if it is not whitelisted. It is also assumed that MAC spoofing is not possible for a device connected over WLAN if different pre-shared keys are used for each device. This is because the device cannot send packets with a different MAC address unless it also knows the corresponding key. However, as discussed above, there is a possibility that wired clients, which were not considered in the approach, could defeat the filtering mechanisms by spoofing their MAC addresses. Assuming that most devices are connected wirelessly (this is likely due to the fact that Smart Home Devices are intended to be placed all over the house and is also indicated by looking at the devices tested by other researchers, such as those listed in ([17], Table II)), a *medium* influence rating is assigned here (*INL*, *RNL*).

MAC, MAH, URC, USC:

While in principle there is no direct influence on the vulnerability of missing access control or authorisation checks, *Bark* can still prevent the attack if the relevant communication is not whitelisted, as discussed in the previous paragraph. The same applies to the input of credentials, whether stolen or guessed. It is assumed that there is no need to access a service (such as SSH or Telnet) if the device is not allowed to log in or perform actions that are not checked. Therefore, a *high* influence rating is assigned (*MAC*, *MAH*, *URC*, *USC*).

MAS:

Obviously, a filtering approach deployed in the home network has *no or only low* influence on missing authorisation checks on the server (*MAS*).

MME:

For the first sub-vector, DNS spoofing, no influence is assumed because in this case the router itself would be the target of the attack and there is no restriction on its own communication that is considered in the filtering approach. Similarly, no influence is assumed in the case of an attacker with physical access to a router or a link on the route. In the case of an Evil Twin, the fact that the clients all use different pre-shared keys makes it impossible for someone who only knows their own password to set up a protected access point that can communicate with any of the other devices. However, it is still possible to simply provide an open access point, even if some devices refuse to connect to it. Therefore, only some influence can be considered for this sub-vector. Note, if a strong end-to-end encryption is additionallyused, including the authentication of the end point to prevent a man-in-the-middle attack on the encrypted end-to-end channel (e.g., https connections with added mechanisms like certificate pinning [92]), then facilitating such an encrypted communication channel can help to protect against MME and MML (see below). Overall, the influence of filtering is rated as *no or only low* for this man-in-the-middle attack (*MME*).

MML:

The influence on MAC address spoofing has already been discussed. For ARP spoofing [95], this means that wired clients can spoof all other clients, while their wireless counterparts can only spoof the devices to which they have access. In any case, if an attacker sends malicious ARP replies to a victim that should not be accessible, the victim may not be able to send messages back to the attacker if that communication is also not allowed.

Furthermore, the attacker may not have permission to forward the messages themselves (e.g., if the victim has access to server A but the attacker does not). Therefore, even in the case of a successful ARP spoofing attack, an attacker may not be able to become a man in the middle. Overall, the impact of this sub-vector is considered to be medium, although the effectiveness depends highly on the scenario, as just discussed. While the situation is not as clear for ARP spoofing, it is much easier for DHCP spoofing. Since the DHCP server is on the home router itself, there is no need to give any client permission to send DHCP solicitations. Therefore, it is assumed that there is a *high* impact on this sub-vector. The effect on Evil Twin attacks has already been rated as *some* influence in the *MME* paragraph above. Overall, the effect for this vector is rated as *medium* (*MML*). Note, if a strong end-to-end encryption alongside an authentication of the endpoint would additionally be facilitated, this would prevent a man-in-the-middle attack (see also above for MME). However, filtering cannot fully protect against MML.

RCE:

Although there is no direct influence on *RCE* vulnerability, *Bark* can still prevent the attack if the corresponding communication is not whitelisted, as previously discussed. If a device is protected, it depends on whether or not the attacking device needs access to the service that also provides the vulnerabilities. As some attacks are prevented while others are not, a *medium* influence rating is assigned here (*RCE*).

SMU:

It is clear that there is *no or only low* influence on the usage of insufficient authentication mechanisms by the server (*SMU*).

SSC:

In the event that an external attacker does not spoof the IP address, their address will not be whitelisted. Nevertheless, this scenario is only applicable if the device is accessible via an open port on the router. In the case where the attacker also spoofs the server's address, the influence is assumed to be *no or only low* according to the discussion on the *INE* vector. If the attacker originates from within the network, the success of the attack is dependent on the communication that has been whitelisted. Overall, a rating of *some* influence is assigned in this instance (*SSC*).

WDA:

Because a WLAN deauthentication attack is performed from outside the local network, *no or only low* influence of a filtering approach is assumed here (*WDA*).

XNE:

All sub-vectors of this vector have been assessed as *no or only low*, which consequently is also the rating for this one (*XNE*).

XNL:

All sub-vectors of this vector have been assessed as *medium*, which consequently is also the rating for this one (*XNL*).

Scenario-Specific Adjustments

S_α (Communication with the Internet):

By definition, communication with the *Unessential Server* is not necessary (see Section 5), and it would therefore not be whitelisted in *Bark*. Because communication with the *Unessential Server* is inhibited, it is assumed that the server stores no or at least fewer data of a device. Hence, the influence on the *DBO* vector is rated as *medium* for the *Unessential Server*. Additionally, because there is no communication from the *Smart Home Device* to this server, it cannot be eavesdropped on, leading to a *high* influence on the corresponding *ENE* vector (This would also apply to some of the other network traffic vectors, which, however, do not appear in a relevant entry).

S_β (Local Network Communication):

According to the definition of scenario S_β in Section 5, the *Smart Home Devices* in the local network are not obligated to communicate with each other, which means that no rules exist that permit any traffic other than that which is originating from the *Owner Control Device* to the *Smart Home Device X*. Because the former is explicitly not considered as an attack device, the influence levels of the local attack vectors *DNL*, *ENL*, *INL*, *MML*, *MNL*, *RNL* and *XNL* are increased to *high* for traffic from or to *Smart Home Device X*.

S_γ (Guest Access):

Similar to S_β, there is no need for communication between the two *Smart Home Devices* according to the specification of Scenario S_γ (see Section 5). Because the *Accessible Smart Home Device* is intended to be accessed by the *Guest Control Device*, there is a need for rules to be in place which allow communication between the two appliances. On the other hand, the *Private Smart Home Device* should by definition not be accessed by the guest, and therefore no rules permitting any communication between them would be in place in *Bark*.

Overall, it is assumed that out of the entities in the local network, *Bark* would only allow the *Owner Control Device* to communicate with the *Private Smart Home Device*. Because the owner has been explicitly excluded as a threat origin, the influence levels of the local attack vectors *DNL*, *ENL*, *INL*, *MML*, *MNL*, *RNL* and *XNL* are increased to *high* for traffic from or to the *Private Smart Home Device*.

S_δ (Indirect Control):

Following the assumptions for this scenario (Section 5), the *Smart Home Device* can only be controlled indirectly. Thus, there is no need for local communication between this device and the *Smart Hub*. Consequently, there would not be any need for filtering (e.g., no need to whitelist local communication), and hence all attack vectors which are solely based on local communication (i.e., *RNL* and *DNL*, *RNL* and *ENL*, *RNL* and *INL*, *RNL* and *MML*, *RNL* and *MNL*, *RNL* and *RNL*, and *RNL* and *XNL*) are influenced at a *high* level. Moreover, the influence on the *DOS* vector is increased to *medium* due to the absence of internal attacks and, in contrast to the previous scenarios, no reflections using the *Owner Control Device* are possible. However, the possibility of an external attack remains.

7.2. Influence of Filtering on STRIDE Impacts

Based on the previous assessments, influences on the threat impacts are aggregated conservatively, i.e., they are aggregated to show the least possible impact by always taking the minimum influence that was assessed for any of the attack vectors that lead to that impact. The results are presented in the Tables 11–14.

Table 11. Influence of *Bark* on STRIDE impacts for S_α. Rating the possible improvement through the usage of filtering: no/only low (grey), medium (yellow) and high (green).

	Spoofing	Tampering	Repudiation	ID	DoS	EoP
Smart Home Device (SHD)	DWA	MFU RCD	SDS SDR	EPI	DNA DLF	
Essential Server (ES)	SWA			SCD(ES)	SNA	IDC
Unessential Server (US)				SCD(US)		NRO
SHD → ES				SCD(SHD→ES)		
SHD → US				SCD(SHD→US)		

Table 12. Influence of *Bark* on STRIDE impacts for S_β. Rating the possible improvement through the usage of filtering: no/only low (grey), medium (yellow) and high (green).

	Spoofing	Tampering	Repudiation	ID	DoS	EoP
Owner Control Device (OCD)	WSI(OCD)	RCD(OCD)		EPI(OCD)	CLF	
Smart Home Device (SHD) *X*	DWA	MFU RCD(SHD)	SDS SDR	EPI(SHD) OCR	DNA DLF	
Server (S) *X*	SWA			SCD(S)	SNA	IDC

OCD → Internet		DEP(OCD→I)		CDD(OCD→I)
				CDM(OCD→I)		

Internet → OCD		DEP(I→OCD)		CDD(I→OCD)
				CDM(I→OCD)		

OCD → SHD				LLM(OCD→SHD)		
SHD → OCD				DCD LLM(SHD→OCD)		
SHD → S				SCD(SHD→S)		

Table 13. Influence of *Bark* on STRIDE impacts for S_γ. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Spoofing	Tampering	Repudiation	ID	DoS	EoP
Owner Control Device (OCD)	WSI(OCD)	RCD(OCD)		EPI(OCD)	CLF(OCD)	
Guest Control Device (GCD)	WSI(GCD)	RCD(GCD)		EPI(GCD)	CLF(GCD)	

Private SHD	DWA(PSHD)	RCD(PSHD)	SDS(PSHD)
			SDR(PSHD)
			GDS(PSHD)
				EPI(PSHD)
				OCR(PSHD)
					DNA(PSHD)
					DLF
						RAC(PSHD)
						GAP

Accessible SHD	DWA(ASHD)	RCD(ASHD)	SDS(ASHD)
			SDR(ASHD)
			GDS(ASHD)
				EPI(ASHD)
				OCR(ASHD)
					DNA(ASHD)
					DLF
						RAC(ASHD)

OCD → Internet		DEP(OCD→I)		CDD(OCD→I)
				CDM(OCD→I)		

Internet → OCD		DEP(I→OCD)		CDD(I→OCD)
				CDM(I→OCD)		

GCD → Internet		DEP(GCD→I)		CDD(GCD→I)
				CDM(GCD→I)		

Internet → GCD		DEP(I→GCD)		CDD(I→GCD)
				CDM(I→GCD)		

OCD → PSHD				LLM(OCD→PSHD)		

PSHD → OCD				DCD(PSHD→OCD)
				LLM(PSHD→OCD)		

OCD → ASHD				LLM(OCD→ASHD)		

ASHD → OCD				DCD(ASHD→OCD)
				LLM(ASHD→OCD)		

GCD → ASHD				LLM(GCD→ASHD)		

ASHD → GCD				DCD(ASHD→GCD)
				LLM(ASHD→GCD)		

Table 14. Influence of *Bark* on STRIDE impacts for S_δ. Rating the possible improvement through the usage of filtering: no/only low (grey), medium (yellow) and high (green).

	Spoofing	Tampering	Repudiation	ID	DoS	EoP

Smart Home Device (SHD)	DWA	MFU
		RCD(SHD)
			SDS(SHD)
			SDR(SHD)
				EPI(SHD)
				SID
					DNA
					DLF	

Smart Hub (SH)	WSI(SH)	RCD(SH) CCR	SDR(SH)	OCR		IDC(SH)
Control Server (CS)	SWA CDR	LCT		SCD(CS)	SNA LCD	IDC(CS)
SHD → CS				SCD(SHD→CS)		
SH → CS				SCD(SH→CS)		

There are only a few threat impacts for which *some* influence was found. These are the threats of wrong actions performed by a *Smart Home Device* (*DWA*), as well as the loss of functionality (*DLF*) of such appliances in S_γ, for which *no or only low* influence was found in the other scenarios (*DWA*, *DLF*). This discrepancy can be explained by considering that for S_γ, no data flows from *Smart Home Devices* to *External Servers* are modelled. Consequently, the vectors with *no or only low* influence ratings (*XNE*, *DOS*), which this communication or the servers themselves introduce in the other scenarios, are therefore omitted in this one.

A *medium* influence was found for extracting private information (*EPI*) from a *Smart Home Device* or running commands to damage it or turn it into a bot (*RCD*). For control devices, even a *high* influence on those vectors was found.

Another point where *Bark* is superior is in the case of protecting the *Private Smart Home Device* from being accessed by the guest (*GAP*), where a *high* influence was found. Unlike all other approaches, the scenario-specific adjustments lead to different influence levels for data collected by the server (*SCD*) between the *Essential Server* and the *Unessential server* in scenario S_α. These adjustments also lead to higher ratings regarding the *Private Smart Home Device* on the threat impacts of obtaining credentials or tokens used for authentication of the user (*OCR*), as well as obtaining (*DCD*) or logging (*LLM*) messages as compared to the *Accessible Smart Home Device*. Those examples show that the consideration of different types of devices and servers changes not only the influence on attack vectors but also the influence on threat impacts.

Another *high* influence was found for changing the control routines on the *Smart Hub* (*CCR*) and a *medium* one for obtaining control over the *Smart Home Device* indirectly by controlling the *Smart Hub* (*IDC(SH)*). Despite those rather positive points, for many entries, *no or only low* influence was identified.

Noticeably, the results show, that *no or only low* influence exists on *Repudiation* threat impacts.

7.3. *Influence of Filtering on LINDDUN Attack Vectors*

After the effect on security threats has been assessed, now, the influence on the privacy-related vectors according to Table A3 is discussed following the same principles as before. Because the effectiveness of filtering-based approaches depends highly on the rules that are in place to filter traffic, scenario-specific circumstances need to be considered. Therefore, scenario-specific adjustments of the influence ratings are discussed after the vector-specific discussions.

BCC, BCS, BCT, BDC:

It is clear that any technical measure taken by a homeowner has no effect on the manner in which a manufacturer designs and builds its products. Therefore, there is no impact on the collection of data on the device. Although a filtering approach may potentially influence the transmission of data, this is not assumed to be the case for data sent by a control or companion app, given that such an app would typically run on a device, such as a Smartphone or tablet, which requires access to the entire Internet. Therefore, the influence on these vectors is rated as *no or only low* (*BCC*, *BCS*, *BCT*, *BDC*).

BDS, BDT, BST:

It can be observed that there is still *no or only low* influence on communication with servers that contribute to the functionality of the device. This is based on the assumption that the functionality of a device should be fully preserved. Nevertheless, an owner may determine that it is preferable to trade some features of the appliance (e.g., remote control) for enhanced privacy (*BDS*, *BDT*, *BST*).

CBL:

The way *Bark* handles broadcast and multicast is to permit transmission to all hosts in the network and to allow them to answer for a specified period of time as indicated by an example of an SSDP discovery ([26], Figure 8). That way an attacker can still collect messages. But this is only true if the sending of such messages is permitted in the first place. Because there is no general discussion about the handling of broadcast and multicast messages given in the paper, based on the given example, the influence on this vector is rated as *some* (*CBL*).

CME:

A rating of *no or only low* influence has been assigned for the *MME* vector on which *CME* is based. However, because of the fact that some communication of the device can be blocked, this leads to a different communication pattern, which may mislead a classification algorithm or lead to uncertainty between devices with similar patterns. Despite that, the filtering may have an influence on the timing of messages. Hence, a rating of *some* influence is assigned (*CME*).

CML:

A *medium* influence rating has been assigned for the *MML* vector on which *CML* is based. Because the attacker may not be able to be an MitM for all traffic, but only regarding the whitelisted one, it can also perceive a different communication pattern, which may mislead a classification algorithm or lead to uncertainty between devices with similar patterns, just as with an external attacker. However, as this does not apply to Evil Twin [96] attacks and broadcast messages can be seen by all devices, it seems unreasonable to assume a high influence on this vector, and therefore, only a *medium* rating is assigned (*CML*).

CMM:

An attacker can still see the MAC address, and hence determine the manufacturer of the device. However, the network behaviour of the device most probably is different in case some traffic to local or external entities is blocked, potentially leading to retransmissions and delayed other messages. Therefore, at least a rating of *some* influence is assigned to this vector (*CMM*).

DCN:

The process of filtering has no direct effect on the user's awareness of the data collected. However, the manner in which Bark presents its rules in a human-readable format may contribute to an understanding of the services with which a product communicates. Furthermore, the reduction in data sent to unessential servers may also result in a smaller discrepancy between the communicated data and the user's expectations.

In general, a rating of at least *some* influence is assigned here (*DCN*). However, this does not apply to the case, where the guest is the victim and the homeowner is collecting the data (*GCH*), perhaps on the home router. Consequently, the influence is rated as *no or only low* there.

HSD:

A homeowner can always install monitoring tools on the home router, independently of the presence of a filtering approach. Therefore, a rating of *no or only low* influence is assigned for this vector (*HSD*).

HUC:

A filtering approach does not create any awareness about any logging activities in the network, so *no or only low* influence is considered for this vector (*HUC*).

ICM:

This vector is simply a collection of the two vectors *CME* and *CML*, for which a rating of *some* and *medium* influence has been assigned, respectively. The minimum influence, which is *some* is considered appropriate here (*ICM*).

IDN:

While the approach cannot change the name of a device, it might prevent others from becoming aware of it by blocking communication which would reveal the name. However, such names can also be part of broadcast or multicast traffic, e.g., in DHCP requests. Because *Bark* seems to not control this type of communication in a fine-grained way, the effect on this sub-vector has to be considered low. On the other hand, a reverse lookup might not be possible if a device can only contact the DNS server with a whitelisted set of domain names. Therefore, there is still a rating of *some* influence assigned here (*IDN*).

IDS:

This vector refers to the identified influence of *Bark* on the corresponding *Information Disclosure* impact in the STRIDE assessment in Tables 11–14. In case multiple impacts exist for the combination of element and threat, the smallest impact is taken. Therefore, no generic influence rating is given here.

IUI:

This vector is based on the ability to know the user of a control device that is given by the impact *IUD*. The vector is further based on the ability to monitor messages in the first place, which is expressed by the corresponding *IDS* vector. Because both of them are necessary, the maximum is taken here. Overall, no generic influence can be given here.

MRD:

This vector refers to a set of *IDS* vectors. Some of the corresponding STRIDE threats might not have been considered relevant, and therefore, there is no entry for them in the STRIDE (Tables 11–14). Consequently, only those vectors, which have a counterpart in the tables are considered. Thus, no general influence rating can be given here.

NSL:

If the rules are sufficiently detailed, they permit only such communication as is necessary. Consequently, an attacker is unable to discover any new hosts or ports by scanning the network. However, the scanning process may still reveal further information, such as the operating system that is running on a host. This does not apply to wired clients, which were not considered in the approach, and which can circumvent the filtering mechanisms by spoofing their MAC addresses. Under the assumption that most devices are wirelessly connected, the influence is rated as *medium* here (*NSL*).

Scenario-Specific Adjustments

S_α (Communication with the Internet):

Because the communication to the *Unessential Server* is by definition not necessary (see Section 5.1), it would not be whitelisted in *Bark*. Additionally, data which have never arrived at the server side can also also not be shared with any other party. Therefore, in the case of the *Unessential Server*, a *high* influence rating is assigned for the vectors *BDS*, *BDT* and *BST*.

S_β (Local Network Communication):

According to the definition of scenario S_β in Section 5.2, the *Smart Home Devices* in the local network do not have to communicate with each other, which means that there would be no rules in place, which permit any traffic other than that which is originating from the *Owner Control Device* to the *Smart Home Device X*. Because the *Owner Control Device* is explicitly not considered as a malicious device, the influence levels of the local attack vector *CML* is increased to *high* for the *Smart Home Device X* as well as traffic from or to it.

S_γ (Guest Access):

Just like in S_β, there is no need for communication between the two *Smart Home Devices* according to the specification of S_γ. Because the *Accessible Smart Home Device* is intended to be accessed by the *Guest Control Device*, there need to be rules in place that allow communication between the two appliances. On the other hand, the *Private Smart Home Device* should by definition not be accessed by the guest, and therefore, no rules permitting any communication between them would be in place in *Bark*. In general, it is assumed that out of the devices in the local network, *Bark* would only allow the *Owner Control Device* to communicate with the *Private Smart Home Device*. Because the *Owner Control Device* has been explicitly excluded as a threat origin, the influence level on the local attack vector *CML* is increased to *high* for the *Private Smart Home Device*, as well as traffic from or to it.

S_δ (Indirect Control):

Because in Section 5.4 it is assumed that the *Smart Home Device* can only be controlled indirectly, there is no need for local communication between this device and the *Smart Hub*. Consequently, there would not be any whitelisted local communication in *Bark*, and therefore, all attack vectors which are solely based on local communication (i.e., *CBL*, *CML*, *IDN* and *NSL*) are influenced at a *high* level.

7.4. Influence of Filtering on LINDDUN Impacts

Just like in the assessments before, influences on the threat impacts are calculated by always taking the minimum influence that was identified for any of the vectors that enable it. The summarised results are shown in Tables 15–18.

Table 15. Influence of *Bark* on LINDDUN impacts for S_α. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Linkability	Identifiability	Detectability	Unawareness	Noncompliance
Smart Home Device (SHD)	LDD	IUD	INA IUA	UCD	MDD

Essential Server (ES)	LDS(ES)	IUS(ES)			MDS(ES)
					MDT(ES)

Unessential Server (US)	LDS(US)	IUS(US)			MDS(US)
					MDT(US)

SHD → ES	LMD(SHD→ES)				
SHD → US	LMD(SHD→US)				

Table 16. Influence of *Bark* on LINDDUN impacts for S_β. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Linkability	Identifiability	Detectability	Unawareness	Noncompliance

Owner Control Device (OCD)	LDC	IUD(OCD)	INA(OCD)
			IUA(OCD)
				UCC	MDD(OCD)

Smart Home Device (SHD) *X*	LDD	IUD(SHD)	INA(SHD)
			IUA(SHD)
				UCD	MDD(SHD)

Server (S) *X*	LDS	IUS			MDS MDT
OCD → Internet	LMD(OCD→I)	IMC(OCD→I)			
Internet → OCD	LMD(I→OCD)	IMC(I→OCD)			
OCD → SHD	LMD(OCD→SHD)	IMC(OCD→SHD)			
SHD → OCD	LMD(SHD→OCD)	IMC(SHD→OCD)			
SHD → S	LMD(SHD→S)				

Table 17. Influence of *Bark* on LINDDUN impacts for S_γ. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red), medium (yellow) and high (green).

	Linkability	**Identifiability**	**Detectability**	**Unawareness**	**Noncompliance**

Owner Control Device (OCD)	LDC(OCD)	IUD(OCD)	INA(OCD)
			IUA(OCD)
				UCC(OCD)	MDD(OCD)

Guest Control Device (GCD)	LDC(GCD)	IUD(GCD)	INA(GCD)
			IUA(GCD)
				UCC(GCD)
				GCH
					MDD(GCD)
					HMD

Private SHD	LDD(PSHD)	IUD(PSHD)	INA(PSHD)
			IUA(PSHD)
				UCD(PSHD)	MDD(PSHD)

Accessible SHD	LDD(ASHD)	IUD(ASHD)	INA(ASHD)
			IUA(ASHD)
				UCD(ASHD)	MDD(ASHD)

OCD → Internet	LMD(OCD→I)	IMC(OCD→I)			MDS MDT
Internet → OCD	LMD(I→OCD)	IMC(I→OCD)			
GCD → Internet	LMD(GCD→I)	IMC(GCD→I)			MDS MDT
Internet → GCD	LMD(I→GCD)	IMC(I→GCD)			
OCD → PSHD	LMD(OCD→PSHD)	IMC(OCD→PSHD)			
PSHD → OCD	LMD(PSHD→OCD)	IMC(PSHD→OCD)			
OCD → ASHD	LMD(OCD→ASHD)	IMC(OCD→ASHD)			
ASHD → OCD	LMD(ASHD→OCD)	IMC(ASHD→OCD)			
GCD → ASHD	LMD(GCD→ASHD)	IMC(GCD→ASHD)			
ASHD → GCD	LMD(ASHD→GCD)	IMC(ASHD→GCD)			

Table 18. Influence of *Bark* on LINDDUN impacts for S_δ. Rating the possible improvement through the usage of filtering: no/only low (grey), some (red) and medium (yellow).

	Linkability	**Identifiability**	**Detectability**	**Unawareness**	**Noncompliance**

Smart Home Device (SHD)	LDD	IUD(SHD)	INA(SHD)
			IUA(SHD)
				UCD	MDD

Smart Hub (SH)	LDH	IUD(SH)	INA(SH)
			IUA(SH)
				UCH	MDH

Control Server (CS)	LDS	IUS			MDS MDT
SHD → CS	LMD(SHD→CS)				
SH → CS	LMD(SH→CS)				

When looking at the tables, it can be seen that there is a higher coverage of entries with at least *some* influence in comparison with the results of the STRIDE assessment. A *medium* influence has been found for linking data on the *Smart Home Device* (*LDD*) for all scenarios except the last one. This can be explained by the fact that for S_δ, the impact of obtaining status information about the *Smart Home Device* from the *Smart Hub* (*SID*), which has a *no or only low* rating, is also considered in the STRIDE assessment for *Information Disclosure*, which in turn determines the *IDS* vector on which the *LDD* impact is based. A *high* influence has been identified for linking data on control devices (*LDC*). Like in the STRIDE assessment, the scenario-specific adjustments lead to higher influence levels between the *Essential Server* and the *Unessential server* in scenario S_α. More precisely, a *medium* influence was found for linking data on the server (*LDS(US)*) or identifying whom these data belong to (*IUS(US)*), and even a *high* influence on the threat, that the server collects or passes on more information than the user agreed on (*MDS(US)*, *MDT(US)*), while *no or only low* influence was found for all of them regarding the *Essential Server*.

8. Conclusions and Future Work

In this paper, we explore the impact of filtering on the security and privacy of IoT systems. As part of the theoretical work, we discuss what a Smart Home and its devices are and what types of devices exist, and we generalise the communication patterns between them. While we extend our discussion to filtering in general, our understanding of the maturity of filtering is that of a central device in a packet-switched network, capable of inspecting the data flow and, based on policies, rejecting any adversarial or unwanted packets. To make this concrete, we have described the filtering technology called the *Bark* approach, which we selected as a model for the filtering capabilities for our impact analysis.

Proceeding to the analysis, the paper identifies four common scenarios in the Smart Home domain and their communication patterns. We believe that these four scenarios can be used either individually or in combination to describe most of today's Smart Home setups. For each scenario, we have performed in-depth STRIDE and LINDDUN analyses. We identify the following: (a) which elements of the system are being targeted, (b) which are potential attack vectors targeting those elements and (c) what the impacts are. This work provides the academic community with 29 security attack vectors, 32 security and privacy impacts, and 20 privacy attack vectors. All of them are systematically identified and presented in tables, which depict the dependencies between impacts and enabling attack vectors for each entity or data flow separately.

Further, we determined that there is a large impact of a filtering method—as powerful as the *Bark* approach—on the identified attack vectors and their impacts. A four-step scale was chosen for evaluation: *no or only low*, *some*, *medium* or *high* impact. This is even though the aggregated impact is conservatively calculated: the overall assessment is marked as the minimum impact of all attack vectors leading to it. So even if two of three attack vectors leading to that impact can be fully mitigated, one can only be mitigated with *some* impact, and the overall impact is rated as *some*. For example, from the STRIDE analysis of an Internet-connected device (see Table 3), when the attacker tries to tamper with the Smart Home Device with the impact to run commands on the device to damage it or turn it into a bot (impact 'RCD'), then even if filtering login attempts to remove two attack vectors (URC, USC), the attack vector of having a remote code execution vulnerability in an interface (attack vector RCE) might only be reduced as some interface might need to remain reachable for functionality.

Figure 12 summarises the analysis of the impact of filtering for the STRIDE and LINDDUN impacts. It is important to note that this does not consider any differences in the severity or likelihood of different threats. Moreover, for comparison, the influence of using a Virtual Private Network (VPN) or an Intrusion Detection System (IDSys) are also included. However, the details for VPN and IDSys had to be left out for brevity.

From the comparison, we can see that Virtual Private Networks (VPNs), at least when installed on the home router, have *no or only low* influence on most of the impacts and *some* influence on only a few attack vectors. They should thus not be suggested or marketed as suitable for maintaining security and privacy in a Smart Home. On the other hand, Intrusion Detection Systems were found to have an effect on many threats, especially in security (STRIDE), while the coverage in the area of privacy (LINDDUN) is not as high. It is important that not a single Intrusion Detection System, but intrusion detection as a whole, was assessed, with results from different approaches. In addition, no distinction was made between detection and prevention. Therefore, those results for IDS are not fully comparable to the ones of filtering, which is fully concerned with the prevention of attacks.

While *Bark* is just a single approach in the filtering-based domain, as it contains basic functionality to filter inbound and outbound directions as well as local and remote devices (see Section 2.4), we deem our observations to be valid for network filtering in general. To check this, we have contrasted the Smart Home in this work with a very different smart device, a modern car. Technically, the filtering—as showcased by existing works [39,40]—also enables in this domain a fine-grained ingress and egress filtering of messages between the vehicle's inside and the outside that can withhold or mask the real values and as such limit the information that leaves the car through an OBD dongle.

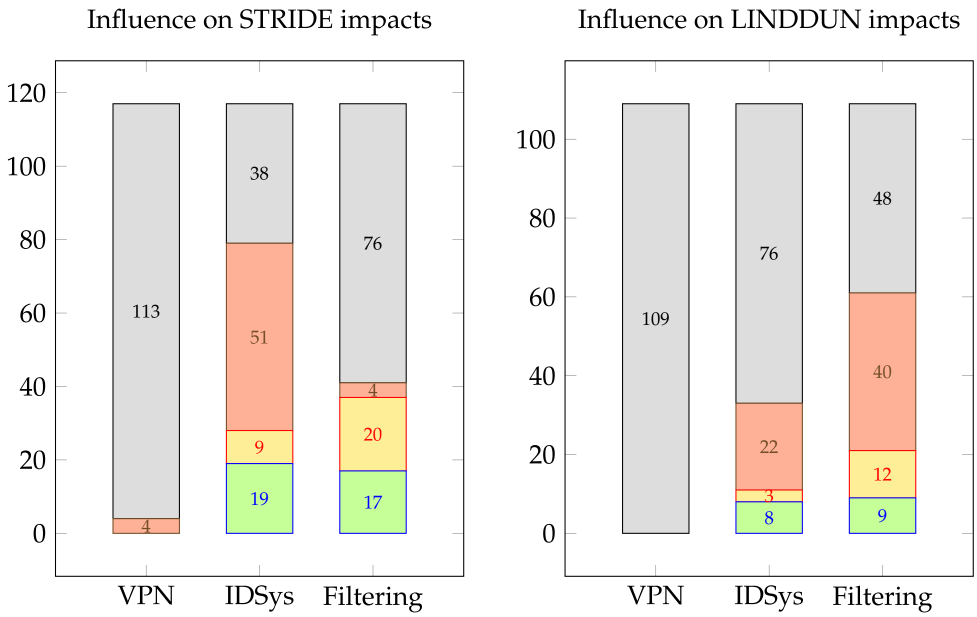

Figure 12. The number of influenced impacts per influence level across all four scenarios for Virtual Private Networks (VPNs), an Intrusion Detection System (IDSys) and filtering using *Bark*. Rating the possible improvement: no/only low (grey), some (red), medium (yellow) and high (green).

Looking at the results in Figure 12 for both an Intrusion Detection System (IDSys) and filtering (*Bark*), it is visible that there are many vectors and threat impacts for which *no or only low* or only *some* influence has been found. This suggests that many threats cannot or not sufficiently be targeted by network security approaches in the Smart Home and require security and privacy by design incorporated into Smart Home Devices (e.g., if data sent or received by the control device are disclosed due to missing encryption (*CDD*)) and deployed on external servers (e.g., to mitigate that a server makes wrong actions (SWA)). However, the results also show that filtering has a *medium* or even *high* influence on threats such as the extraction of private information from a device (*EPI*) or that someone executes malicious commands on a device to turn it into a bot or damage it (*RCD*). Neither Intrusion Detection Systems (IDSys) nor filtering dominates the other regarding every threat category. However, privacy filtering overall shows a larger influence than the other approaches.

Overall, the results suggest that filtering-based approaches cannot prevent all threats but have an impact on many of the threats that arise in the Smart Home.

Future Work

Since this topic is quite broad, many potential directions for future work can be identified; we believe that the following ones may be the most interesting:

- **Improve the STRIDE and LINDDUN analysis**: Taking the weight of the different threat impacts or attack vectors into account when determining the influence of the filtering was out of our scope. Therefore, future work could extend the existing set of impacts with such weights, e.g., through a risk assessment or user study.

- **Improve the filtering rule-set generation**: Developing a filtering approach that allows for broad coverage of Smart Home Devices and their usage in everyday situations. We have briefly touched on some ideas on how to assist the end user in generating rules in Section 2.5. This could be further extended. As mentioned, the user could be assisted by allowing more dynamic, as e.g., proposed in [47], and more playful interactions with the rule set, leading to potentially temporarily bounded rule-set and thus hopefully a further reduction in threats and their impacts, for example, allowing devices only to connect to specific servers at specific times to download updates, instead of potentially pulling updates from anywhere on the Internet. This can be achieved through improved flexibility in visualisation for the manual configuration of filtering rules or by automatically generating filtering rules by the constant observation of user behaviour. Further, an automated recognition of device types, as e.g., proposed in [17], could be a desirable functionality to automatically infer the type of a new device. Rules could also be provided by a third party, allowing users to download them, facilitating a yet-to-be-designed modular rule package format. Additionally, rule generation can be improved by using AI-based Intrusion Detection Systems [46,97] or by using automated tools that can check if data flows or device's network capabilities comply with predefined rules, such as *COPSEC* checking for GDPR compliance [44]. Then, rules can be created based on the results of those systems or the vulnerabilities found. This would further enhance the reliability of filtering and provide additional assistance to the user in making decisions regarding filtering rules.
- **Improve networks' ability to separate data flows**: Successful network filtering requires that the filtering device is in a unique position of the network in order to see and thus filter the data flow. Configuring the magnitude of devices, e.g., providing them the Smart Home's WLAN passphrase requires often tedious manufacturer setup procedures. In this work, we only very briefly touched on the aspect that the 802.11 WLAN standard [33] would enable to use the passphrase different for each wireless device. We have termed this idea as *per-device-802.11-passphrase* to specify and look up if a different key for the encryption and decryption is within the 802.11 WLAN standard, which specifies explicitly a 'keyLookup' [33] operator and is implemented within the Linux implementation of a WLAN access point `hostapd`. (The documentation from the `hostapd` package (see https://packages.debian.org/en/sid/hostapd (accessed: 19 January 2025)) states that 'Optionally, WPA PSKs can be read from a separate text file (containing list of (PSK, MAC address) pairs.' Note that this list often is formatted to be first MAC then PSK.) The latter would separate the devices cryptographically and not only prevent direct communication among the wireless clients but also means that one compromised device would not leak a common passphrase to the WLAN. In combination with per-device filtering rules, this would also limit the effect of MAC spoofing: an attacker could communicate in the way filtering allows it for that spoofed device but not more. Further, the proliferation of software-defined networks (SDNs) into also home networks in the future would further allow to make use of the facilities within software-defined networks to enforce rules to limit the data flow [47,97], extract the data flow for visualisation [47], or use it for identifying problems using an AI-based intrusion detection [97].
- **Improve the coverage and decrease the load of security functionality provided by the network**: If devices themselves are insecure or too weak to uphold the needed level of security, e.g., they obtain no updates but are vulnerable, other devices within a Smart Home IoT network could take over. The weaker or insecure devices could delegate their filtering (or other security functions) to other Smart Home Devices.

Author Contributions: Conceptualization, H.C.P., F.K. (Fabian Kügler), E.G. and F.K. (Felix Klement); Methodology, H.C.P., F.K. (Fabian Kügler), E.G. and F.K. (Felix Klement); Validation, H.C.P., F.K. (Fabian Kügler), E.G. and F.K. (Felix Klement); Formal analysis, H.C.P., F.K. (Fabian Kügler), E.G. and F.K. (Felix Klement); Investigation, H.C.P., F.K. (Fabian Kügler), E.G. and F.K. (Felix Klement); Writing—original draft, H.C.P., F.K. (Fabian Kügler), E.G. and F.K. (Felix Klement); Writing—review & editing, H.C.P., F.K. (Fabian Kügler), E.G. and F.K. (Felix Klement). All authors have read and agreed to the published version of the manuscript.

Funding: The work of Felix Klement, Emiliia Geloczi and Henrich C. Pöhls was funded by the Bavarian State Ministry of Science and Arts (BayStMWK), under the projects 'Secure Encapsulation' and 'Architectures and services for secure everyday digitalisation' of the Bavarian Research Association 'FORDaySec' (see https://fordaysec.de (accessed on 19 January 2025)): 'Security in Everyday Use of Digital Technologies'. The work of Fabian Kügler has been carried out during his Master Thesis at the University of Passau.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Abbreviations

Table A1. The attack vector descriptions along with their abbreviations for STRIDE.

ARC	Attacker logs into the server with randomly guessed admin credentials
ASC	Attacker logs into the server with stolen admin credentials
DBO	Database stores information open (i.e., accessible from outside and not protected or only weakly protected)
DIL	There is no or no sufficient logging of messages or acknowledgements from others on the device or in the network
DNE	Attacker outside the local network drops traffic using an MitM position (MME)
DNL	Attacker inside the local network drops traffic using an MitM position (MML)
DOS	Any form of Network Denial of Service attack (e.g., DDoS using SYN-Flooding)
DRC	Attacker tries random credentials or tokens to impersonate the Smart Home Device
DSC	Attacker impersonates the Smart Home Device with stolen credentials
ENE	Attacker outside the local network eavesdrops on traffic using an MitM position (MME)
ENL	Attacker inside the local network eavesdrops on traffic using an MitM position (MML)
INE	Attacker outside the local network inserts traffic with spoofed IP address
INL	Attacker inside the local network inserts traffic with spoofed IP or MAC address
LRC	Attacker tries random credentials to log into a legitimate user account on the server
LSC	Attacker logs into a legitimate user account on the server with stolen credentials
MAC	Missing access control on the Smart Home Device
MAH	Missing authorisation checks on the hub (e.g., when adding rules)
MAS	Missing authorisation checks on the server (e.g., when registering for remote access in an app)
MME	Attacker performs a man-in-the-middle attack outside the local network (DNS spoofing, physical access to a router or wire on the route (e.g., national agency or Internet Service Provider), Evil Twin)
MML	Attacker performs a man-in-the-middle attack inside the local network (ARP spoofing, DHCP spoofing, Evil Twin)
MNE	Attacker outside the local network modifies traffic using an MitM position (MME)

Table A1. *Cont.*

MNL	Attacker inside the local network modifies traffic using an MitM position (MML)
RCE	An interface (e.g., web interface) of the device has a RCE vulnerability
RNE	Attacker outside the local network replays traffic
RNL	Attacker inside the local network replays traffic
SMU	Messages from the server are not authenticated (e.g., with digital signatures, message authentication codes are not sufficient)
SSC	Attacker impersonates the server using stolen credentials (e.g., leaked private key)
URC	Attacker tries random or default credentials to log into one of the device's user accounts (potentially root) (even behind NAT this might be possible if the device installed an open port using uPnP)
USC	Attacker logs into one of the device's user accounts (potentially root) using stolen credentials (even behind NAT this might be possible if the device installed an open port using uPnP)
WDA	Attacker performs a WLAN deauthentication attack on the device
XNE	Attacker interferes with the traffic from outside the local network (DNE, INE, RNE, MNE)
XNL	Attacker interferes with the traffic from inside the local network (DNL, INL, RNL, MNL)

Table A2. The impact descriptions along with their abbreviations for STRIDE.

NR	Not relevant
NC	Not clear if there are threats (other than the ones already covered in different categories)
CCR	Changing the control routines
CDD	Disclosure of the data sent or received by the control device
CDM	Disclosure of the metadata of messages sent or received by the control device (this may e.g., reveal visited web pages)
CDR	Attacker might remotely control the Smart Home Device
CLF	The control device loses Internet-related functionality (which may be most of the functionality e.g., for a Smartphone)
DCD	Depends on the data sent by the Smart Home Device, which might be less severe data like logs concerning the device's functionality or very sensitive data revealing private information or video material
DEP	Depends highly on the usage of the control device or the security level of the websites it connects to (e.g., no Internet access, no file syncing with remote client/server, session fixation/hijacking when browsing, etc.)
DLF	The Smart Home Device loses its functionality (totally or partially)
DNA	The Smart Home Device obtains wrong or no information and therefore does not carry out an action (e.g., unlocking the door or raising an alarm)
DWA	The Smart Home Device obtains wrong information and in turn makes wrong actions (e.g., unlocking the door or raising an alarm)
EPI	Attacker extracts private information from the device
GAP	Guest accesses the Private Smart Home Device
GDS	Guest denies the sending of a message (which might have triggered actions like unlocking the door or raising an alarm)
IDC	Attacker obtains control over the Smart Home Device indirectly by controlling an entity that already has control over it
LCD	A legitimate user (e.g., Smart Hub or Remote Control Device) loses control over the Smart Home Device because the control server is unavailable

Table A2. *Cont.*

LCT	A legitimate user (e.g., Smart Hub or Remote Control Device) loses control over the Smart Home Device because the attacker deleted his or her account information on the Control server
LLM	Attacker can log (potentially encrypted) messages in the local network (e.g., to replay them later on)
MFU	A malicious firmware update is sent to the Smart Home Device
NRO	Not relevant on its own but may lead to other threats
OCR	Attacker obtains credentials or tokens used for authentication of the user
RAC	Guest obtains remote access to the Smart Home Device when only local or no access was intended
RCD	Attacker runs commands on the device to damage it or turn it into a bot
SCD	Depending on the data collected by the server, which might be less severe data like logs concerning the device's functionality or very sensitive data revealing private information or video material
SDR	Server denies receipt of a message (which would have triggered actions like, e.g., buying something or calling the fire department)
SDS	Server denies the sending of a message (which might have triggered actions like unlocking the door or raising an alarm)
SID	Attacker obtains status information about the Smart Home Device from another entity
SNA	The server obtains wrong or no information and therefore does not carry out an action (e.g., buying something or calling the fire department)
SWA	The server obtains wrong information and in turn makes wrong actions (e.g., buying something or calling the fire department)
WSI	The device has wrong status information about a Smart Home Device it controls

Table A3. The attack vector descriptions along with their abbreviations for LINDDUN.

BCC	Manufacturer built the control/companion app in a way such that it collects data it should not
BCS	Manufacturer built the control/companion app in a way such that it sends data it should not send to the server
BCT	Manufacturer built the control/companion app in a way such that it sends data it should not send to a third party
BDC	Manufacturer built the device in a way such that it collects data it should not
BDS	Manufacturer built the device in a way such that it sends data it should not send to the server
BDT	Manufacturer built the device in a way such that it sends data it should not send to a third party
BST	Manufacturer built the server in a way such that it sends data it should not send to a third party
CBL	Attacker collects local broadcast messages which contain information that identifies a device
CME	Attacker outside the local network collects metadata of the network traffic by observing network traffic leaving the home using an MitM position (MME) and compares them to patterns from known devices
CML	Attacker inside the local network collects metadata of the network traffic using an MitM position (MML) and compares them to patterns from known devices
CMM	Attacker outside the local network collects metadata of the network traffic by observing encrypted WLAN traffic and compares them to patterns from known devices
DCN	The user has not been informed about the data collected
HSD	homeowner spies on device's activity
HUC	homeowner unaware of data collection/logging enabled in the network

Table A3. *Cont.*

ICM	Indirect CME or CML in which the traffic of the Smart Hub indicates the device it is controlling
IDN	The device has an identifying device name (e.g., Tim's iPhone)
IDS	Information disclosure of the corresponding entry in the STRIDE modelling
IUI	The user is identified by linking the source or destination address with the corresponding device's user
MRD	The messages sent by or sent to the device are readable (IDS(d → x), IDS(x → d)) and contain identifiers such as an email address or physical properties of a person
NSL	Attacker scans the internal network (e.g., using arp or pings, perhaps by using a tool like nmap) for available hosts or open ports and services offered by a host.

Table A4. The impact descriptions along with their abbreviations for LINDDUN.

NR	Not relevant
NC	Not clear if there are threats (other than the ones already covered in different categories)
GCH	Guest is unaware of the data or metadata collected/logged by the homeowner
HMD	There is more information collected/logged by the homeowner than the guest agreed on
IMC	Identify the user based on messages sent to or by the control device.
INA	Inventory attack (i.e., the attacker gets to know that there is a device of the type of the Smart Home Device at home)
IUA	Infer a user's activity (e.g., when a user goes to bed)
IUD	Identify the user of the device (e.g., when an identifier like an email address or physical properties of a person is stored on the device)
IUS	Identify who's data is stored on the server (e.g., when an identifier like an email address or physical properties of a person is stored on the server together with usage data)
LDC	Link data on the control device (because of data probably stored in user-specific folders on a multi-user PC or only one user on single-user devices like Smartphones, this will in most cases be trivial)
LDD	Link data on the Smart Home Device (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)
LDH	Link data on the Smart Hub (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)
LDS	Link data on the server (e.g., entries with different timestamps whenever a user unlocks the door to create a profile of when the house is occupied)
LMD	Link messages of the device (e.g., at different points in time to create a profile of when the user is asleep)
MDD	The device itself collects more information than the user agreed on or is reasonable
MDH	The Smart Hub collects or sends more information than the user agreed on or is reasonable (e.g., about the usage of devices under control to the manufacturer of the hub)
MDS	The server collects more information than the user agreed on or is reasonable
MDT	There is more information passed to third parties than the user agreed on or is reasonable
NRO	Not relevant on its own but may lead to other threats
UCC	User is unaware of the data sent by the control/companion app of the Smart Home Device
UCD	User is unaware of the data collected or sent by the Smart Home Device
UCH	User is unaware of the data collected or sent by the Smart Hub

References

1. Plume. Plume IQ 1H 2022 Smart Home Market Report. 2022 Available online: https://plumestrong.plume.com/1h-report/p/1 (accessed on 30 November 2024).
2. Joshi, S. 70 IoT Statistics to Unveil the Past, Present, and Future of IoT. 2023. Available online: https://learn.g2.com/{IoT}-statistics (accessed on 15 June 2023).
3. Medaglia, C.M.; Serbanati, A. An Overview of Privacy and Security Issues in the Internet of Things. In *The Internet of Things*; Giusto, D., Iera, A., Morabito, G., Atzori, L., Eds.; Springer: New York, NY, USA, 2010; pp. 389–395.
4. Raymond, J.F. Traffic analysis: Protocols, attacks, design issues, and open problems. In *Proceedings of the Designing Privacy Enhancing Technologies*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 10–29.
5. Danezis, G.; Clayton, R. Introducing traffic analysis. In *Digital Privacy: Theory, Technologies, and Practices*; Auerbach Publications: Boca Raton, FL, USA, 2007; pp. 1–24.
6. Staudemeyer, R.C.; Pöhls, H.C.; Wójcik, M. What it takes to boost Internet of Things privacy beyond encryption with unobservable communication: A survey and lessons learned from the first implementation of DC-net. *J. Reliab. Intell. Environ.* **2019**, *5*, 41–64. [CrossRef]
7. Staudemeyer, R.C.; Pöhls, H.C.; Wójcik, M. The road to privacy in IoT: Beyond encryption and signatures, towards unobservable communication. In Proceedings of the 7th workshop on IoT-SoS: Internet of Things Smart Objects and Services (WOWMOM SOS-IOT 2018), Chania, Greece, 12–15 July 2018. [CrossRef]
8. Buil-Gil, D.; Kemp, S.; Kuenzel, S.; Coventry, L.; Zakhary, S.; Tilley, D.; Nicholson, J. The digital harms of Smart Home Devices: A systematic literature review. *Comput. Hum. Behav.* **2023**, *145*, 107770. [CrossRef]
9. Hnatyuk, K. Internet of Things (IoT) Statistics: 2022/2023. 2023. Available online: https://marketsplash.com/Internet-of-things-statistics (accessed on 29 November 2024).
10. Souppaya, M.; Montgomery, D.; Polk, T.; Ranganathan, M.; Dodson, D.; Barker, W.; Johnson, S.; Kadam, A.; Pratt, C.; Thakore, D.; et al. *Securing Small-Business and Home Internet of Things (IoT) Devices: Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD)*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2021. [CrossRef]
11. Pöhls, H.C.; Petschkuhn, B.; Rückert, J.; Mössinger, M. Aggregation and Perturbation in Practice: Case-Study of Privacy, Accuracy and Performance. In Proceedings of the 19th IEEE International Workshop on Computer Aided Modelling Analysis and Design of Communication Links and Networks (CAMAD 2014), Athens, Greece, 1–3 December 2014. [CrossRef]
12. Elicegui, I.; Carrasco, J.; Escribano, C.P.; Gato, J.; Becerra, A.; Politis, A., Usage-Based Automotive Insurance. In *Big Data and Artificial Intelligence in Digital Finance: Increasing Personalization and Trust in Digital Finance Using Big Data and AI*; Springer International Publishing: Cham, Switzerland, 2022; pp. 295–311. [CrossRef]
13. Gram-Hanssen, K.; Darby, S.J. "Home is where the smart is"? Evaluating Smart Home research and approaches against the concept of home. *Energy Res. Soc. Sci.* **2018**, *37*, 94–101. [CrossRef]
14. Bugeja, J.; Jacobsson, A.; Davidsson, P. On Privacy and Security Challenges in Smart Connected Homes. In Proceedings of the 2016 European Intelligence and Security Informatics Conference (EISIC), Uppsala, Sweden, 17–19 August 2016; pp. 172–175. [CrossRef]
15. Sivaraman, V.; Gharakheili, H.H.; Vishwanath, A.; Boreli, R.; Mehani, O. Network-level security and privacy control for smart-home IoT devices. In Proceedings of the 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Abu Dhabi, United Arab Emirates, 19–21 October 2015; pp. 163–167. [CrossRef]
16. Apthorpe, N.; Reisman, D.; Feamster, N. Closing the Blinds: Four Strategies for Protecting Smart Home Privacy from Network Observers. *arXiv* **2017**, arXiv:1705.06809.
17. Miettinen, M.; Marchal, S.; Hafeez, I.; Asokan, N.; Sadeghi, A.; Tarkoma, S. IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 2177–2184. [CrossRef]
18. Apthorpe, N.; Reisman, D.; Feamster, N. A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic. *arXiv* **2017**, arXiv:1705.06805.
19. Schiefer, M. Smart Home Definition and Security Threats. In Proceedings of the 2015 Ninth International Conference on IT Security Incident Management IT Forensics, Magdeburg, Germany, 18–20 May 2015; pp. 114–118. [CrossRef]
20. *ISO/IEC 27403*; (FDIS Final Draft) Cybersecurity—IoT Security and Privacy—Guidelines for IoT-Domotics. ISO: Geneve, Switzerland, 2024.
21. *ISO/IEC 27400*; Cybersecurity—IoT Security and Privacy—Guidelines. ISO: Geneve, Switzerland, 2022.
22. Greer, C.; Burns, M.J.; Wollman, D.A.; Griffor, E.R. *Cyber-Physical Systems and Internet of Things*; Special Publication (NIST SP); National Institute of Standards and Technology: Gaithersburg, MD, USA, 2019. [CrossRef]
23. Whitmore, A.; Agarwal, A.; Da Xu, L. The Internet of Things—A survey of topics and trends. *Inf. Syst. Front.* **2015**, *17*, 261–274. [CrossRef]
24. Barrera, D.; Molloy, I.; Huang, H. IDIoT: Securing the Internet of Things like it's 1994. *arXiv* **2017**, arXiv:1712.03623.

25. Notra, S.; Siddiqi, M.; Habibi Gharakheili, H.; Sivaraman, V.; Boreli, R. An experimental study of security and privacy risks with emerging household appliances. In Proceedings of the 2014 IEEE Conference on Communications and Network Security, San Francisco, CA, USA, 29–31 October 2014; pp. 79–84. [CrossRef]
26. Hong, J.; Levy, A.; Riliskis, L.; Levis, P. Don't Talk Unless I Say So! Securing the Internet of Things with Default-Off Networking. In Proceedings of the 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), Orlando, FL, USA, 17–20 April 2018; pp. 117–128. [CrossRef]
27. Geismann, J.; Bodden, E. A systematic literature review of model-driven security engineering for cyber–physical systems. *J. Syst. Softw.* **2020**, *169*, 110697. [CrossRef]
28. Shostack, A. *Threat Modelling: Designing for Security*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
29. Kohnfelder, L.; Garg, P. The Threats to Our Products. April 1999. Available online: https://shostack.org/files/microsoft/The-Threats-To-Our-Products.docx (accessed on 19 January 2025).
30. Khan, R.; McLaughlin, K.; Laverty, D.; Sezer, S. STRIDE-based threat modelling for cyber-physical systems. In Proceedings of the 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Turin, Italy, 26–29 September 2017; pp. 1–6. [CrossRef]
31. 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.* **2011**, *16*, 3–32. [CrossRef]
32. Roberts, L. The Evolution of Packet Switching. *Proc. IEEE* **1978**, *66*, 1307–1313. [CrossRef]
33. *IEEE 802.11*; Wireless LAN Medium Access Control and Physical Layer Specifications. IEEE: Piscataway, NJ, USA, 1999.
34. Bellovin, S.M. Distributed Firewalls. In *;login:*; November 1999; Issue 11-1999, pp. 37–39. Available online: https://www.cs.columbia.edu/~smb/papers/distfw.pdf (accessed on 19 January 2025).
35. Stefanopoulou, A.; Dimara, A.; Michailidis, I.; Karatzinis, G.; Papaioannou, A.; Krinidis, S.; Anagnostopoulos, C.N.; Kosmatopoulos, E.; Ioannidis, D.; Tzovaras, D. Ensuring Reliability in Smart Building IoT Operations Through Real-Time Holistic Data Treatment. In Proceedings of the Artificial Intelligence Applications and Innovations, AIAI 2023 IFIP WG 12.5 International Workshops, León, Spain, 14–17 June 2023; Maglogiannis, I., Iliadis, L., Papaleonidas, A., Chochliouros, I., Eds.; Springer: Cham, Switzerland, 2023; pp. 207–218.
36. Pöhls, H.C.; Karwe, M. Redactable Signatures to Control the Maximum Noise for Differential Privacy in the Smart Grid. In Proceedings of the 2nd Workshop on Smart Grid Security (SmartGridSec 2014), Munich, Germany, 26 February 2014; Lecture Notes in Computer Science (LNCS); Cuellar, J., Ed.; Springer International Publishing: Cham, Switzerland, 2014; Volume 8448. Available online: http://link.springer.com/chapter/10.1007/978-3-319-10329-7_6/fulltext.html (accessed on 19 January 2025).
37. Day, J.D.; Zimmermann, H. The OSI reference model. *Proc. IEEE* **1983**, *71*, 1334–1340. [CrossRef]
38. Klement, F.; Pöhls, H.C.; Spielvogel, K. Towards Privacy-Preserving Local Monitoring and Evaluation of Network Traffic from IoT Devices and Corresponding Mobile Phone Applications. In Proceedings of the IEEE 3rd Workshop on Internet of Things Security and Privacy (WISP 2020) Held in Conjunction with Global IoT Summit 2020 (GIOTS 2020), Dublin, Ireland, 3–5 June 2020. [CrossRef]
39. Klement, F.; Pöhls, H.C.; Katzenbeisser, S. Man-in-the-OBD: A modular, protocol agnostic firewall for automotive dongles to enhance privacy and security. In Proceedings of the 5th International Workshop on Attacks and Defenses for Internet-of-Things (ADIoT 2022) in Conjunction with ESORICS, Copenhagen, Denmark, 30 September 2022; Lecture Notes in Computer Science (LNCS); Volume 13745, pp. 143–164. [CrossRef]
40. Klement, F.; Pöhls, H.C.; Katzenbeisser, S. Change Your Car's Filters: Efficient Concurrent and Multi-Stage Firewall for OBD-II Network Traffic. In Proceedings of the 2022 IEEE 27th International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), Paris, France, 2–3 November 2022; pp. 19–25. [CrossRef]
41. Bellovin, S.M.; Cheswick, W.R. Network firewalls. *IEEE Commun. Mag.* **1994**, *32*, 50–57. [CrossRef]
42. Sanders, C. *Practical Packet Analysis, 3E: Using Wireshark to Solve Real-World Network Problems*; No Starch Press: San Francisco, CA, USA, 2017.
43. Serror, M.; Henze, M.; Hack, S.; Schuba, M.; Wehrle, K. Towards In-Network Security for Smart Homes. In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES 2018), Hamburg, Germany, 27–30 August 2018; pp. 18:1–18:8. [CrossRef]
44. Anselmi, G.; Mandalari, A.M.; Lazzaro, S.; De Angelis, V. COPSEC: Compliance-Oriented IoT Security and Privacy Evaluation Framework. In Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), Madrid, Spain, 2–6 October 2023. [CrossRef]
45. European Parliament and the Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union, L 119. 4 May 2016, pp. 1–88. Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=oj:JOL_2016_119_R_TOC (accessed on 19 January 2025).

46. Sowmya, T.; Mary Anita, E. A comprehensive review of AI based Intrusion Detection System. *Meas. Sens.* **2023**, *28*, 100827. [CrossRef]
47. Rakotondravony, N.; Pöhls, H.C.; Pfeifer, J.; Harrison, L. Viz4NetSec: Visualizing Dynamic Network Security Configurations of Everyday Interconnected Objects in Home Networks. In Proceedings of the HCI for Cybersecurity, Privacy and Trust: 6th International Conference, HCI-CPT 2024, Held as Part of the 26th HCI International Conference, HCII 2024, Washington, DC, USA, 29 June–4 July 2024; Proceedings, Part II; LNCS; pp. 164–185. [CrossRef]
48. Geloczi, E.; Pöhls, H.C.; Klement, F.; Posegga, J.; Katzenbeisser, S. Unveiling the Shadows: An Approach towards Detection, Precise Localization, and Effective Isolation of Concealed IoT Devices in Unfamiliar Environments. In Proceedings of the 23rd Workshop on Privacy in the Electronic Society (WPES '24), Salt Lake City, UT, USA, 14–18 October 2024; pp. 109–123. [CrossRef]
49. Lyu, M.; Lau, L. Firewall security: Policies, testing and performance evaluation. In Proceedings of the 24th Annual International Computer Software and Applications Conference (COMPSAC2000), Taipei, Taiwan, 25–27 October 2000; pp. 116–121. [CrossRef]
50. Hamed, H.; Al-Shaer, E. Dynamic rule-ordering optimization for high-speed firewall filtering. In Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security (ASIACCS '06), Taipei, Taiwan, 21–24 March 2006; pp. 332–342. [CrossRef]
51. Molina-Markham, A.; Shenoy, P.; Fu, K.; Cecchet, E.; Irwin, D. Private Memoirs of a Smart Meter. In Proceedings of the 2nd ACM BuildSys '10, Zurich, Switzerland, 2 November 2010; pp. 61–66. [CrossRef]
52. Enev, M.; Gupta, S.; Kohno, T.; Patel, S.N. Televisions, video privacy, and powerline electromagnetic interference. In Proceedings of the ACM CCS, Chicago, IL, USA, 17–21 October 2011; pp. 537–550. [CrossRef]
53. Jacobsson, A.; Boldt, M.; Carlsson, B. On the Risk Exposure of Smart Home Automation Systems. In Proceedings of the 2014 International Conference on Future Internet of Things and Cloud, Barcelona, Spain, 27–29 August 2014; pp. 183–190. [CrossRef]
54. Jacobsson, A.; Boldt, M.; Carlsson, B. A risk analysis of a Smart Home automation system. *Future Gener. Comput. Syst.* **2016**, *56*, 719–733. [CrossRef]
55. Geneiatakis, D.; Kounelis, I.; Neisse, R.; Nai-Fovino, I.; Steri, G.; Baldini, G. Security and privacy issues for an IoT based Smart Home. In Proceedings of the 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 22–26 May 2017; pp. 1292–1297. [CrossRef]
56. Heartfield, R.; Loukas, G.; Budimir, S.; Bezemskij, A.; Fontaine, J.R.; Filippoupolitis, A.; Roesch, E. A taxonomy of cyber-physical threats and impact in the Smart Home. *Comput. Secur.* **2018**, *78*, 398–428. [CrossRef]
57. De Donno, M.; Dragoni, N.; Giaretta, A.; Mazzara, M. AntibIoTic: Protecting IoT Devices Against DDoS Attacks. In Proceedings of the 5th International Conference in Software Engineering for Defence Applications, Rome, Italy, 7–8 June 2018; Ciancarini, P., Litvinov, S., Messina, A., Sillitti, A., Succi, G., Eds.; Springer: Cham, Switzerland, 2018; pp. 59–72. [CrossRef]
58. Ghiglieri, M.; Waidner, M. HbbTV Security and Privacy: Issues and Challenges. *IEEE Secur. Priv.* **2016**, *14*, 61–67. [CrossRef]
59. Gebhardt, J.; Massoth, M.; Weber, S.; Wiens, T. Ubiquitous Smart Home control on a Raspberry Pi embedded system. In Proceedings of the UBICOMM 2014—8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Rome, Italy, 24–28 August 2014; pp. 172–177. Available online: https://www.thinkmind.org/articles/ubicomm_2014_6_30_10109.pdf (accessed on 19 January 2025).
60. Haar, C.; Buchmann, E. FANE: A Firewall Appliance for the Smart Home. In Proceedings of the 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), Leipzig, Germany, 1–4 September 2019; pp. 449–458. [CrossRef]
61. European Parliament and the Council of the European Union. Regulation (EC) No. 715/2007 of the European Parliament and of the Council of 20 June 2007 on Type Approval of Motor Vehicles with Respect to Emissions from Light Passenger and Commercial Vehicles (Euro 5 and Euro 6) and on Access to Vehicle Repair and Maintenance Information. 2007. Available online: https://www.legislation.gov.uk/eur/2007/715 (accessed on 19 January 2025).
62. Robert Bosch GmbH. *CAN Specification*; Version 2.0; Robert Bosch GmbH: Stuttgart, Germany, 1991. Available online: http://esd.cs.ucr.edu/webres/can20.pdf (accessed on 19 January 2025).
63. *Standard ISO 11898-1*; Road Vehicles—Controller Area Network (CAN)—Part 1: Data Link Layer and Physical Signalling. International Organization for Standardization: Geneva, Switzerland, 2015.
64. Wen, H.; Chen, Q.A.; Lin, Z. Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A New Over-the-Air Attack Surface in Automotive IoT. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Online, 12–14 August 2020; pp. 949–965.
65. Yadav, A.; Bose, G.; Bhange, R.; Kapoor, K.; Iyenger, N.C.S.N.; Caytiles, R. Security, Vulnerability and Protection of Vehicular On-board Diagnostics. *Int. J. Secur. Its Appl.* **2016**, *10*, 405–422. [CrossRef]
66. Keegan, J.; Ng, A. Who Is Collecting Data from Your Car? 2022. Available online: https://themarkup.org/the-breakdown/2022/07/27/who-is-collecting-data-from-your-car (accessed on 19 January 2025).
67. El Basiouni El Masri, A.; Artail, H.; Akkary, H. Toward self-policing: Detecting drunk driving behaviours through sampling CAN bus data. In Proceedings of the 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, United Arab Emirates, 21–23 November 2017; pp. 1–5. [CrossRef]

68. Nirmali, B.; Wickramasinghe, S.; Munasinghe, T.; Amalraj, C.R.J.; Bandara, H.M.N.D. Vehicular data acquisition and analytics system for real-time driver behaviour monitoring and anomaly detection. In Proceedings of the 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 15–16 December 2017; pp. 1–6. [CrossRef]
69. Srinivasan, A. IoT Cloud Based Real Time Automobile Monitoring System. In Proceedings of the 2018 3rd IEEE International Conference on Intelligent Transportation Engineering, Singapore, 3–5 September 2018; pp. 231–235. [CrossRef]
70. Bernardini, C.; Asghar, M.R.; Crispo, B. Security and privacy in vehicular communications: Challenges and opportunities. *Veh. Commun.* **2017**, *10*, 13–28. [CrossRef]
71. Hoppe, T.; Kiltz, S.; Dittmann, J. Security Threats to Automotive CAN Networks—Practical Examples and Selected Short-Term Countermeasures. In Proceedings of the 27th International Conference on Computer Safety, Reliability, and Security (SAFECOMP '08), Newcastle upon Tyne, UK, 22–25 September 2008; pp. 235–248. [CrossRef]
72. Studnia, I.; Nicomette, V.; Alata, E.; Deswarte, Y.; Kaâniche, M.; Laarouchi, Y. Security of embedded automotive networks: State of the art and a research proposal. In Proceedings of the SAFECOMP 2013—Workshop CARS (2nd Workshop on Critical Automotive Applications: Robustness & Safety) of the 32nd International Conference on Computer Safety, Reliability and Security, Toulouse, France, 24–27 September 2013.
73. Wolf, M.; Weimerskirch, A.; Paar, C. Security in Automotive Bus Systems. 2004. Available online: https://api.semanticscholar.org/CorpusID:16502503 (accessed on 19 December 2024).
74. Semiconductors, N. Automotive Gateway: A Key Component to Securing the Connected Car. 2018. Available online: https://www.nxp.com/docs/en/white-paper/AUTOGWDEVWPUS.pdf (accessed on 19 December 2024).
75. Robert Bosch GmbH. Bosch Central Gateway. 2022. Available online: www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/connectivity-solutions/central-gateway/ (accessed on 19 December 2024).
76. Karamba Security. 2016. Available online: https://karambasecurity.com (accessed on 19 December 2024).
77. Rizvi, S.; Willett, J.; Perino, D.; Vasbinder, T.; Marasco, S. Protecting an Automobile Network Using Distributed Firewall System. In Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing (ICC '17), Cambridge, UK, 22–23 March 2017. [CrossRef]
78. The European Parliament and the Council of the European Union. Directive 1995/46/EC of the European Parliament and of the Council—On the Protection of Individuals with Regard to the Processing of Personal Data on the Free Movement of Such Data. Official Journal of the European Union, L 281. 23 November 1995, pp. 31–50. Available online: https://eur-lex.europa.eu/eli/dir/1995/46/oj/eng (accessed on 19 January 2025).
79. EU Article 29 Data Protection Working Party (WP 223). Opinion 8/2014 on the Recent Developments on the Internet of Things. 16 September 2014. Available online: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp223_en.pdf (accessed on 19 January 2025).
80. European Union Agency for Cybersecurity; Domingo-Ferrer, J.; Hansen, M.; Hoepman, J.; Le Métayer, D.; Tirtea, R.; Schiffner, S.; Danezis, G. Privacy and Data Protection by Design—From Policy to Engineering. December 2014. Available online: https://data.europa.eu/doi/10.2824/38623 (accessed on 19 January 2025).
81. Ontario. Office of the Information and Privacy Commissioner; Cavoukian, A. Privacy by Design—The 7 Foundational Principles—Implementation and Mapping of Fair Information Practices. Toronto, ON, Canada, 2009. Available online: https://privacy.ucsc.edu/resources/privacy-by-design---foundational-principles.pdf (accessed on 19 January 2025).
82. Kung, A.; Kargl, F.; Suppan, S.; Cuellar, J.; Pöhls, H.C.; Kapovits, A.; McDonnell, N.N.; Martin, Y.S. A privacy engineering framework for the Internet of things. In *Data Protection and Privacy: (In) Visibilities and Infrastructures*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 163–202. [CrossRef]
83. Ernst & Young. Data Loss Prevention Keeping Your Sensitive Data out of the Public Domain. Insights on Governance, Risk and Compliance. EYG no. AU0969. 2011. Available online: https://cloudsynergyinc.com/assets/pdfs/Data-Loss-Prevention.pdf (accessed on 19 January 2025).
84. Furnell, S.; Thomson, K.L. Recognising and addressing 'security fatigue'. *Comput. Fraud. Secur.* **2009**, *2009*, 7–11. [CrossRef]
85. Cram, W.A.; Proudfoot, J.G.; D'Arcy, J. When enough is enough: Investigating the antecedents and consequences of information security fatigue. *Inf. Syst. J.* **2021**, *31*, 521–549. [CrossRef]
86. Pöhls, H.C.; Rakotondravony, N. Dynamic Consent: Physical Switches and Feedback to Adjust Consent to IoT Data Collection. In Proceedings of the Distributed, Ambient and Pervasive Interactions: 8th International Conference, DAPI 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, 19–24 July 2020; Proceedings 22; pp. 322–335. [CrossRef]
87. Miller, B.; Nixon, T.; Tai, C.; Wood, M. Home networking with Universal Plug and Play. *IEEE Commun. Mag.* **2001**, *39*, 104–109. [CrossRef]
88. Hunt, T. Data from Connected CloudPets Teddy Bears Leaked and Ransomed, Exposing Kids' Voice Messages. 2019. Available online: https://www.troyhunt.com/data-from-connected-cloudpets-teddy-bears-leaked-and-ransomed-exposing-kids-voice-messages/ (accessed on 19 December 2024).

89. Sivaraman, V.; Chan, D.; Earl, D.; Boreli, R. Smart-Phones Attacking Smart-Homes. In Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '16), Darmstadt, Germany, 18–20 July 2016; pp. 195–200. [CrossRef]
90. Conti, M.; Dragoni, N.; Lesyk, V. A Survey of Man In The Middle Attacks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2027–2051. [CrossRef]
91. Amazon.com. Understand the Smart Home Skill API | Alexa Skills Kit. Available online: https://developer.amazon.com/de/docs/smarthome/understand-the-smart-home-skill-api.html (accessed on 19 December 2024).
92. Evans, C.; Palmer, C.; Sleevi, R. Public Key Pinning Extension for HTTP. RFC 7469. 2015. Available online: https://doi.org/10.17487/RFC7469 (accessed on 19 December 2024).
93. Cheng, H.; Avnur, R. *Traffic Analysis of SSL Encrypted Web Browsing*; Project Paper; University of Berkeley: Berkeley, CA, USA, 1998.
94. Lyu, M.; Sherratt, D.; Sivanathan, A.; Gharakheili, H.H.; Radford, A.; Sivaraman, V. Quantifying the Reflective DDoS Attack Capability of Household IoT Devices. In Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '17), Boston, MA, USA, 18–20 July 2017; pp. 46–51. [CrossRef]
95. Bai, X.; Hu, L.; Song, Z.; Chen, F.; Zhao, K. Defense against DNS Man-In-The-Middle Spoofing. In *Proceedings of the Web Information Systems and Mining*; Gong, Z., Luo, X., Chen, J., Lei, J., Wang, F.L., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 312–319.
96. Bauer, K.; Gonzales, H.; McCoy, D. Mitigating Evil Twin Attacks in 802.11. In Proceedings of the 2008 IEEE International Performance, Computing and Communications Conference, Austin, TX, USA, 7–9 December 2008; pp. 513–516. [CrossRef]
97. Li, J.; Zhao, Z.; Li, R.; Zhang, H. AI-Based Two-Stage Intrusion Detection for Software Defined IoT Networks. *IEEE Internet Things J.* **2019**, *6*, 2093–2102. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.