information

Communication

Integrating Human Factors in the Visualisation of Usable Transparency for Dynamic Risk Assessment

Anastasija Collen [1,*], Ioan-Cosmin Szanto [2], Meriem Benyahya [1], Bela Genge [3] and Niels Alexander Nijdam [1]

[1] Centre Universitaire d'Informatique, Geneva School of Economics and Management, University of Geneva, Route de Drize 7, CH-1227 Carouge, Switzerland; meriem.benyahya@unige.ch (M.B.); niels.nijdam@unige.ch (N.A.N.)
[2] Kalos Information Systems, 540142 Targu Mures, Romania ; szanto.cosmin@gmail.com
[3] Department of Electrical Engineering and Information Technology, Faculty of Engineering and Information Technology, 'George Emil Palade' University of Medicine, Pharmacy, Science and Technology of Târgu Mureș, 540142 Targu Mures, Romania; bela.genge@umfst.ro
* Correspondence: anastasija.collen@unige.ch

Abstract: Modern technology and the digitisation era accelerated the pace of data generation and collection for various purposes. The orchestration of such data is a daily challenge faced by even experienced professional users in the context of Internet of Things (IoT)-enabled environments, especially when it comes to cybersecurity and privacy risks. This article presents the application of a user-centric process for the visualisation of automated decision making security interventions. The user interface (UI) development was guided by iterative feedback collection from user studies on the visualisation of a dynamic risk assessment (DRA)-based security solution for regular lay users. The methodology we applied starts with the definition of the methodological process to map possible technical actions to related usable actions. The definition and refinement of the user interface (UI) was controlled by the survey feedback loop from end user studies on their general technological knowledge, experience with smart homes, cybersecurity awareness and privacy preservation needs. We continuously improved the visualisation interfaces for configuring a cybersecurity solution and adjusting usable transparency of the control and monitoring of the dynamic risk assessment (DRA). For this purpose, we have designed, developed and validated a decision tree workflow and showed the evolution of the interfaces through various stages of the real-life trials executed under European H2020 project GHOST.

Keywords: usable security; IoT; smart home; security; privacy; risk assessment; user-centric development

Citation: Collen, A.; Szanto, I.-C.; Benyahya, M.; Genge, B.; Nijdam, N.A. Integrating Human Factors in the Visualisation of Usable Transparency for Dynamic Risk Assessment. *Information* **2022**, *13*, 340. https://doi.org/10.3390/info13070340

Academic Editors: Enrico Denti and Corinna Schmitt

Received: 12 April 2022
Accepted: 10 July 2022
Published: 14 July 2022

1. Introduction

The IoT is a powerful emerging technology that has been developed to make the home environment smarter and more secure, connected and automated [1], although the technologies supporting smart home functionalities ushered in new daunting cybersecurity and privacy challenges [2]. While security became one of the first priorities in software development, a multitude of challenges were identified by developers experiencing difficulties in integrating security principles into the designs and structures of their implementations [3]. Consequently, an eager need for tools providing visibility of cyber risks and threats has been raised in parallel with the deployment of cutting-edge smart homes [4]. For that matter, dynamic risk assessment (DRA)-based tools are foreseen to allow smart home users to take control and make appropriate decisions regarding the existing cybersecurity and privacy risks [5]. Such technology intends to automate threat identification and provide control and monitoring features for mitigation any detected risks. However, DRA's prevalence depends on how its user interface (UI) is tweaked based on user feedback and involvement.

Professional cybersecurity analytics tools have evolved tremendously in the last decade. Previously, many tools were developed for security analysts but mainly contained tables and text, being intended for professional use only. However, recent advancements demonstrate that data visualisation tools for information retrieval (IR) have completely changed security, as they enable users and analysts to understand information about their network security more easily [6].

Nevertheless, the usability aspects are often neglected. Poor usability of cybersecurity solutions tends to be the effect of security constraints. Finding the right trade-off between usability and security or preferably integrating the usability and security requirements is part of a major research challenge which has recently been raised by scholars [7]. For instance, user-centred approaches are recommended as a means to accomplish usable security [8], while definition of the objectives for both security and usability is suggested as a way to decide on the right balance between the two [9]. Understanding security and usability collectively is recognised as a critical factor for the successful development, implementation and usage of an information system (ISys), according to Andriotis et al. [10]. As far as the IoT is concerned, usability is among the major research challenges identified [11]. Consecutively, we have observed a growth in privacy concerns as IoT device manufacturers for the smart home are acquired by large corporations such as Google [12]. The most recent research suggests new usable security frameworks, particularly for modelling security and privacy risks in smart homes at the consumer level. For example, the framework presented in [13] aims to support home users with a highly usable security decision support tool. However, it still needs to address improvements in usability and scalability and validate the real utility offered to the user.

Existing cybersecurity solutions tend to provide increased protection at the expense of usability [9,14], a choice that typically backfires in practice because of end user demotivation due to poor usability, leading to even weaker protection. On the contrary, our implementation targets achieving a substantial increase in usability with minimal security trade-offs. To realise this ambition, we have adopted a threefold strategy that builds upon extensive automation (minimising security-related user interactions), user motivation and building trust.

Our work was performed under the umbrella of the GHOST project, offering a smart home cybersecurity software solution (https://cordis.europa.eu/project/id/740923, accessed on 31 March 2022). The GHOST solution is a cybersecurity framework built up from a series of software modules, each with their distinctive functionality, forming an execution chain consisting of analysing network traffic, detecting anomalies, determining risks, identifying the threats, providing mitigation strategies, performing automated decisions and ultimately informing and offering control to the end user [15]. Within the GHOST project, deployment of the solution was conducted in several countries directly to home users. Through two different manufacturers and service providers, a home gateway was installed upon which the solution was integrated into their proprietary software services. Both gateways were ARM processor-based. One was custom fabricated, and the other was based on a Raspberry Pi 3. The work presented in this paper is part of the last sequence in the execution chain, namely automating mitigation decisions and informing the end user. It therefore directly follows up the research on risk assessment (RA) [5], which serves as an input to this work.

Another noteworthy part of the research inputs are the usability studies [16–18] performed to investigate how to optimally bring security- and privacy-related information, which tends to be technical in nature, to the lay users. These studies focus on the perception of the user towards cybersecurity and privacy risks. The general finding on privacy risk perception is that lay users have a vague understanding due to not being affected in a direct, perceivable manner. Through a scientific approach, we investigated and provided a solution on how to translate and map the technical actions to usable actions. Technical actions are the actions that can be performed to mitigate a problem, whereas usable actions are actions intended to be presented to the user as part of the GHOST solution. As several

technical actions may be automated, depending on the situation, different strategies may be used for mapping to the appropriate usable action.

This article aims to reply to the following research questions:

RQ 1. *What are the limitations to automating technical actions in case of detected risk exposure in the scope of the threat landscape specific to smart home environments?*

RQ 2. *How can these technical actions be translated to the lay user, assuring high usability and efficient cybersecurity?*

RQ 3. *Can we have equally engaged lay users with different security and privacy perception and risk acceptance?*

Our added value and contributions can be summarised as follows:

- Define and demonstrate the methodological process for mapping technical actions to usable and automatable mitigation techniques;
- Test our model in a smart home environment and present the results found based on an interactive user-centric approach;
- Design a decision tree conceptualisation by exploring the perception of the end users regarding security, privacy-related risks and the associated Information Retrieval (IR) methods in the context of usable security;
- Implement the decision tree model as a result of our research and translate it into a final set of decision-making monitor and control user interfaces (UIs).

The remainder of this paper is structured as follows. Section 2 discusses the related work. Section 3 presents the research method, describing the six phases of the 'User Actions Mapping' methodology depicted in Figure 1. Section 4 explores the threat vectors applicable to smart home ecosystems. Section 5 maps the threats to technical actions as required by phase two and three of the methodology. Section 6 incorporates dynamic risk assessment (DRA), as part of the fourth phase, into our design to dynamically evaluate the cybersecurity risks. Section 7 focuses on the two final phases, implementing a decision tree conceptualisation and presenting the interface results. Finally, Section 8 presents an analysis on research questions and highlights the challenges and future orientations, followed by Section 9, which concludes the paper.

2. Related Work

The focus of our work is the development of a reusable framework to map technical risk mitigation actions to usable user interfaces (UIs). However, such mapping implies a deep understanding of all intermediary steps from attack identification and mitigation to usability aspects and human factor inclusion in the design of the User Interface (UI). Therefore, the focus lies on four interrelated fields of research, starting from the existing usability recommendations in the security domain, their applicability in terms of a lay user's risk perception, tailoring those risks into threat consequence mitigation and finally the inclusion of the user in the whole process of developing the final solution.

2.1. Security Usability Guidelines

The foundation work on the guidelines for secure interaction design widely applied in real-life products was provided by Yee [19]. It is grouped into three pillars, each providing more fine-grained recommendations on the design of the interfaces:

- General principles: relies on the path of least resistance and appropriate boundaries;
- Actor-ability state maintenance: achieved by explicit authorisation, visibility, revocability and expected ability;
- Communication with the user: accomplished with a trusted path, identifiability, expressiveness and clarity.

However, usable security offers much more than graphical interfaces. It addresses how people think about and use computer systems. The authors of [20] attempted to analyse three organisational case studies, aiming at improving their security products. They have observed that usability is never a priority for the development of new products, and it was mostly introduced due to customer's complaints. Furthermore, usable security is not seen as a quality improvement property, and its definition differs from one organisation to another. In addition, there are no clearly defined evaluation criteria for usability, and therefore, developers are not capable of delivering usability, as they lack understanding of its impact on the end users' performance. Based on these findings, the authors raise an open question: *'Does risk-based security make security more usable than compliance-based security?'*.

Balfanz et al. [21] provided lessons extracted from their work while building a usable security solution. First, one cannot retrofit usable security; instead, the interaction principles should be changed at the core of the product. This hinders the need for end user inclusion in the whole design and the development process of the security product. Second, developers should be careful about the front-end UI layers of the application design. If a security feature prevents the user from accomplishing a certain task without being exposed to what is happening within the system of their device, then that security feature is likely to be turned off completely due to a lack of comprehension. To that end, transparent and understandable control functionality of the security software solution is highly necessary. Finally, security expertise should not outweigh the end user's needs.

In terms of the usability measurement techniques, Atzeni et al. [22] proposed considering the following characteristics: effectiveness, satisfaction, accuracy, efficiency, memorability and knowledge. However, the authors stressed that such measurements should not be utilised on their own as they stand but in a comparative manner from the improvement perspective. This leads to the need for an iterative design process and thus to continuously improving the usability.

2.2. Security and Privacy Risk Perception

Gerber et al. [17] performed a study on privacy risk perception in three distinct domains: online social networks, smart homes and smart healthcare. Their findings pointed to a common lack of understanding of the consequences of risk exposure. More specifically, they evaluated the differences between perception of the abstract and specific risk scenarios related to cyber risks. A proposition of the design of a risk perception intervention to raise the security and privacy threats attitude, specifically in the smart home environment, was advanced in the follow-up work [18]. Their proposed intervention targets vision enhancement for the smart home as a whole.

Another study was run by Barbosa et al. [23] to understand the smart home device adoption limitation factors. The researchers identified three clusters of consumers. For the first category—affordability-oriented—the primary criteria were the actual prices of the desired devices. The second cluster's representatives—privacy-oriented—prioritised personal data preservation politics. The final group of consumers—reliability-oriented—were guided mostly by the assessment of the devices' functionality. Similarly, Emami-Naeini et al. [24] studied IoT security and privacy labels by utilising a layered approach for information visualisation in the context of smart home device acceptability barriers for lay users. Interestingly, they confirmed the known phenomenon of the *Privacy Paradox*, which is an observation of the discrepancy between privacy concerns and the actions taken to mitigate those concerns.

The end user perception of the IoT data field practices and associated risks was analysed in an interview-based study [25]. The authors claimed that the results indicate differences in how the lay user mentally models risks and protection behaviour, highly depending on their background and experience. This implies the need for designing an adoptable UI to suit the diverse needs and support distinct perceptions of the end users.

More specifically, a set of recommendations is derived from this work, highlighting the need for transparent controls and educating regular citizens about the involved risks.

2.3. Risk Assessment for Threat Mitigation as a Usability Improvement

As outlined by Bugeja et al. [26], the smart home domain is fulfilled by the security- and privacy-related challenges dictated by three conceptual sources: devices, communication channels and services provided. This survey also served as a basis for the derivation of the threat landscape applicable to our methodology, which is further described in Section 4. The authors also stressed the need for empirical risk evaluation methods to facilitate and improve the usability of cybersecurity solutions for lay users.

To have a better understanding of the current threat landscape in IoT-enabled systems, a honeypot-based environment was deployed by researchers to capture the most recent snapshot of the existing risks [27]. The outcome of their 6-month experiment confirms the relevance of traditional cyberattacks in smart home environments.

Interview-based research was accomplished by Haney et al. [28] to identify the currently available and used in practice mitigation techniques for non-technical end users for security and privacy risks in smart home environments. The resulting conclusion clearly demonstrates the lack of available tools even for more technology-savvy users. A set of guidelines was derived from this study, pointing to the emerging need for data collection and cybersecurity transparency, privacy and security controls and general assistance availability for lay users.

2.4. User-Centric Approaches

The experience-centred approach in privacy and security technologies was developed by Dunphy et al. [29]. The value and necessity of the proposed approach are argued to be encouraged by the changing context in which the technology is to be deployed. More specifically, the authors demonstrated three use cases, with each outlining a different approach but permitting the establishment od findings that other methods would not be able to capture:

- Collage building: enables the end user to be in charge of the engagement method and the extent of their contribution and experience sharing;
- Questionable concepts: facilitates the expression of opinions as provocative concepts are proposed by the designers;
- Digital portraits: permits establishing trust between participating parties.

All three methods are applicable at the solution design phase and emphasise a secondary place from the technology itself, spotlighting the desire to use the technology of the lay user.

A plethora of user-centric approaches was proposed afterwards in the academic research in the domain of security, privacy and trust design. Collard and Briggs [30] analysed a range of the relevant tool kits and assessed their effectiveness through the execution of a series of workshops. The methods they utilised were based on story-telling, visual and 3D modelling, improvisation and role-play, games and cards and finally on problem setting and mapping. Each of the applied methods showed potential in discovering versatile results in the design phases. A visualisation of the outgoing network traffic of a smart home through the application of participatory design was presented in a recent work of Victora [31]: *IoTGuard*. While not targeting directly the provision of cybersecurity risk exposure mitigation, *IoTGuard* targets providing transparency and control options to the lay user to improve the understanding of smart home cyber risks.

Awareness campaigns in cybersecurity have raised researchers' attention, with the purpose of studying the influencing factors of online behaviour change for a lay user. The main hypothesis of Bada et al. [32] for failing awareness campaigns relied on the fact that security interfaces, as they are developed, are often too difficult to be used by a lay user. They have proposed generic classification of the influencing factors into two domains: personal factors and cultural and environmental factors. While personal factors

are recognised to be formed by an individual's knowledge, skills, personal motivation and experience, the cultural and environmental factors stem from the collective phenomenon of self-perception. The authors of the same work also identified that techniques used for persuasion of behavioural change rely on versatile elements such as fear, humour, expertise, repetition and scientific evidence. However, none of them were shown to be more effective than the others.

We have identified a general lack of user-centric methods applicable directly to cybersecurity in the smart home domain, especially with the focus on risk assessment (RA). One of the closest works we found was on the topic of user experience (UX) in the design of smart home devices. While not addressing directly the question of providing a methodology for the development of ad hoc cybersecurity solutions for the smart home, the authors suggested guidelines to improve data protection in smart homes through a series of interviews [33]. Their work shows a persisting need for a methodological design process to deliver security solutions for smart home environments.

In the same context, Feth et al. [34] presented a user-centred design model for usable security systems, relying on four iterative pillars:

1. Context of use: defined by the users, tasks and environment;
2. System awareness: ensures the system is understandable for the end user by mapping the conceptual model to the user's mental model;
3. System design: selects UI patterns to support the envisioned functionality of the final system;
4. Design evaluation: an iterative cycle through feedback collection and analysis.

While the proposed framework remains an abstract concept, the authors demonstrate its theoretical application to IoT device deployment in a smart home by a fictitious user.

Finally, a thorough survey was presented by [35] in the domain of human-centric cybersecurity. Their work defines a wide perspective as a generic framework applicable to cybersecurity products encompassing the user, usage and usability (i.e, the 3Us). Such a taxonomy enables efficient positioning of any existing methods in the domain of usable security.

3. Methodology for User Action Mapping

To address the emerging need for usability in cybersecurity, we defined a methodology for the user-centric development of usable user actions to ensure transparent monitoring and control of the detected risks. The main phases of our methodology are depicted in Figure 1.

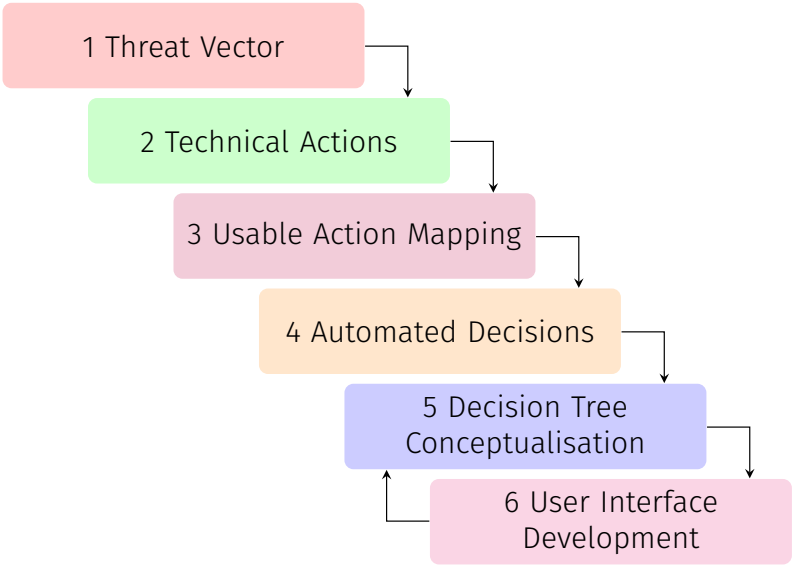

Figure 1. Methodology of User Actions Mapping.

The first phase, the threat vector, is concerned with identifying the relevant threat vectors applicable to the smart home environment. This was performed through interviews and discussions with the manufacturers of the gateways and cybersecurity experts within the GHOST project. Further guided by the deduced threat taxonomy, the operational context and deployment constraints, we established a list of attacks which are technically feasible to simulate in a safe setting without endangering the end user being exposed to real cyber risks (see detailed analysis in Section 4). Table 1 provides a summarised overview of the selected attacks, their demonstration methodologies and the associated tooling for attack replication.

Table 1. Summary of attacks.

Attacks		ID	Validation Methodology	Software Tools
Physical attacks	Physical damage	P1	Remove battery, shut down	N/A
	Malicious device injection	P2	Device registration, sniffers	N/A
	Mechanical exhaustion	P3	Trigger device operation	N/A
Network attacks	Traditional attacks	N1	Scanning and enumeration	nmap, Scapy, tcpreplay
	Device impersonation	N2	Packet injection	Scapy, tcpreplay, tcprewrite
	Side channel attacks	N3	Hardware or software sniffers	Wireshark, tcpdump
	Unusual activities and battery-depleting attacks	N4	Packet injection, sniffers	Scapy, tcpreplay, tcprewrite
Software attacks	Traditional attacks	S1	Traffic replay	PCAP files, tcpreplay, tcprewrite
	Compromised software attacks	S2	Alter module behaviour	Module-specific software
	Command injection	S3	Inject legitimate commands	Specially crafted software
	Mechanical exhaustion	S4	Inject legitimate commands	Specially crafted software
	Sleep deprivation	S5	Inject legitimate commands	Specially crafted software

The second phase was the identification of the technical actions applicable to each category of attacks, which are outlined in detail in Section 5. Those represent the techniques that can be used to address the attacks. Similar to the previous phase, the results were established through interviews and discussions with the manufacturers and cybersecurity experts within the GHOST project. A summary of the possible actions and their potential in automation is provided in Table 2.

Table 2. Technical actions for attacks.

ID	Description	Automatable
T1	Verify physical integrity	No
T2	Verify battery	No
T3	One-way sandboxing	Yes
T4	Two-way sandboxing	Yes
T5	Permit	Yes
T6	Block device temporarily	Yes
T7	Block device permanently	Yes
T8	Drop packets for flow temporarily	Yes
T9	Drop packets for flow permanently	Yes
T10	Drop packets for source temporarily	Yes
T11	Drop packets for source permanently	Yes
T12	Restart GHOST	Yes
T13	Restart module	Yes
T14	Disable module temporarily	Yes
T15	Disable module permanently	Yes
T16	Send update request	Yes

The third phase is the definition of possible usable actions to enable the lay user to control and monitor their smart home from the cybersecurity and risk evolution perspective. In light of the technical capabilities and limitations of the gateway, together with the manufacturers and usability security experts, we assessed a set of possible actions, resulting in five actions that we not only technically could propagate to the end user but also present in a meaningful manner. Table 3 shows the final mapping between the identified usable and technical actions. This mapping also served as a basis for automated decision derivation (fourth phase) and the corresponding conceptualisation of the decision tree (fifth phase). The fourth phase, the automated decisions are a crucial step in providing an actual degree of control for the user, namely by providing the option to resolve an identified risk through an automated solution where possible. The user may decide to be informed of any automated action or to be in full control through what we call the awareness preferences. The fifth phase focuses on the direct interaction of the user with the GHOST solution and establishes a decision tree based on the analysis of the threat vector scenarios and the awareness preferences. The fourth and fifth phases are described in detail in Sections 6 and 7, respectively.

Table 3. Usable actions.

ID	Description	Linked Technical Actions
U1	Allow	T5, T16
U2	Block	T3, T4, T6, T7, T8, T9, T10, T11
U3	Ignore	T3, T4, T5, T6, T7, T8, T9, T10, T11, T16
U4	Remind	T3, T4, T5, T6, T7, T8, T9, T10, T11, T16
U5	Advisory	T1, T2, T12, T13, T14, T15

The sixth and final phase concerns the actual software implementation of the user interfaces, which were continuously validated by the aforementioned user studies running in the scope of the GHOST project. Throughout several iterations, the outlook and presentation were adapted based on the recommendations resulting from the analysis of the user surveys and their feedback collection [16,17]. The outline of the interactive user-centred approach is detailed for two types of interfaces, configuration (CFG) and security intervention (SI), in Sections 7.2 and 7.3, respectively.

4. Threat Vector Landscape

We analysed the existing threat landscape applicable to smart home environments to define an initial set of applicable attacks. The most notable works providing the taxonomy of smart home-specific attacks and threats were provided by Bugeja et al. [26] and Heartfield et al. [36]. With cross-correlation performed in the scope of risk identification under the GHOST project [5], we further reduced the initial listing by applying criteria on the attack simulation feasibility. The sections hereafter outline each category of attacks included in our analysis along with the scenario definition, implemented demonstration and validation methodology.

4.1. Physical Attacks

Per the summary outlined in Table 1, the physical attacks category consists of substantial damage, malicious device injection and mechanical exhaustion threats. The following analyses advance knowledge on the appropriateness of the demonstration methodology, with a focus on each subcategory of physical attacks.

4.1.1. Physical Damage (P1)

Scenario: Physical damage to an IoT device may be caused by various means: removing the battery, shutting down the device, physically breaking the device (the communication component specifically or complete physical destruction), etc. However, irrespective

on the actual cause, the result will be the same: communications between the device and the IoT gateway will be interrupted. An attacker may pursue the effective physical destruction of IoT devices for various reasons. For example, the attacker may break into a house and (physically) disable sensors or actuators in order to avoid alarms being triggered.

Demonstration and validation methodology: According to the specific functioning of each device available at the deployment site, the attack is demonstrated via the following:

- Removing the battery: Z-Wave and Zigbee devices;
- Shutting down the device: Wi-Fi devices.

4.1.2. Malicious Device Injection (P2)

Scenario: The 'injection' of a new device may happen in various scenarios. This might not necessarily represent an attack scenario, since a user may just want to extend the set of IoT devices with new ones. On the other hand, an attacker may attempt to add a device to an existing network, in which case the dynamic risk assessment (DRA) should signal this attempt to the end user. An attacker may pursue the injection of new devices for various reasons. For example, the attacker may wish to indirectly trigger events in other IoT devices (e.g., trigger fire alarms). Conversely, the attacker may want to alter the behaviour of an installation by flooding it with packets or valid requests, which may lead to an effective denial-of-service (DoS) attack by filling the gateway's disk with event logs, flooding the user interface with alerts and valid events.

Demonstration and validation methodology: Two scenarios are demonstrated to replicate the attack:

- Following the typical procedures for adding a new device to the IoT installation;
- Using a Z-Wave sniffer to demonstrate the sniffing of events via a new device.

4.1.3. Mechanical Exhaustion (P3)

Scenario: The mechanical exhaustion implies that an attacker with physical access to an IoT device is able to repeatedly trigger the mechanics of a particular device. For example, in the case of an IoT switch, the attacker may repeatedly turn the device on and off (at a higher rate than expected). The objective of the attack may be diverse, but the attacker may try to cause mechanical exhaustion which, in time, may lead to malfunctioning or physical damage. The attacker may also exploit the physical access to a device to indirectly trigger other devices. Accordingly, in an IoT scenario, it is common to have an event originating from a particular device trigger actions in other devices. Therefore, the attacker may indirectly and repeatedly trigger the other devices as well which, in turn, may also be damaged.

Demonstration and validation methodology: The attack is demonstrated by repeatedly physically triggering an IoT device (e.g., a switch or relay). The triggering needs to be performed at a higher rate than would be expected in the device's normal operation.

4.2. Network Attacks

Shifting from physical threats, the following discussion represents a wide spectrum of network attacks. More specifically, we shed light on the demonstration of traditional network-related attacks, device impersonation and side channel threats, in addition to unusual activities and battery-depleting assaults.

4.2.1. Traditional Attacks (N1)

Scenario: Traditional network attacks imply the exploitation of the operation of traditional IP-based protocols. In this category, we find the well-known network scanning and device enumeration techniques (TCP/IP- and UDP-related scans) and Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks. Given that the smart home is expected to use IP-based protocols to communicate with remote sites, it is therefore important that the Dynamic Risk Assessment (DRA) is aware of traditional attacks that exploit IP-based protocols. The attack recreates the typical steps that would be performed by an

attacker in the attempt to discover IoT devices and gateways and to enumerate the available services.

Demonstration and validation methodology: Traditional tools will be used to mount attacks against the IoT gateway, including the following:

- `nmap`: SYN scan, XMAS scan and full scan;
- `tcpreplay, tcprewrite, Scapy`: DoS attack and Distributed denial-of-service (DDoS) attack.

4.2.2. Device Impersonation (N2)

Scenario: By leveraging specially crafted hardware, an attacker may use a registered device's identifier and key in order to inject packets, trigger other devices and sniff out the communication network. Indeed, the attack requires a higher level of technical knowledge and a deep understanding of the underlying infrastructure and the communication protocols. Furthermore, in case communications are encrypted, the attacker needs to gain possession of the encryption key. This may be possible due to flaws in cryptographic algorithms or cryptographic protocols or due to flawed implementations. This attack will enable device impersonation, and the attacker may attempt to trigger other devices to open doors, trigger alarms and simulate a presence. The attacker may also attempt to gain possession of alerts issued by other devices. To this end, sensitive data such as a user's presence may be revealed by other devices, which may by used to break into the user's house.

Demonstration and validation methodology: The demonstration methodology for this attack will not attempt to recreate the full attack vector. Instead, we emulate a device impersonation by specially forging new data packets (specific to the IoT protocol (e.g., Bluetooth or Z-Wave)) and by injecting these packets directly to the network capture.

4.2.3. Side Channel Attacks (N3)

Scenario: Side channel attacks try to extract information indirectly from the behaviour of a particular system. In the case of network attacks, side channel attacks attempt to infer sensitive data on users and devices based on the network protocol implementation but not its contents. The attack may be used, for example, to infer user presence or particular events (e.g., an alarm triggered or switch triggered). The attack shall evaluate the level of sensitive information that can be inferred by an attacker that passively sniffs and analyses network traffic. An attacker may attempt to infer sensitive user or device information by sniffing communications (which may be encrypted) and by analysing the unencrypted packet headers (data link, IP, TCP/UDP, Z-Wave or Bluetooth). This way, the attacker may learn that the user is not home or that the door or window is open or closed. This information may then be used for breaking into the user's house.

Demonstration and validation methodology: The demonstration focuses on IoT device traffic and gateway traffic, thus covering both IP-based and non IP-based traffic. In particular, for specific protocols, the following tools will be considered:

- Bluetooth: packets may be sniffed by leveraging the device's capability (e.g., Android has a built-in sniffer);
- Z-Wave: a special device is needed to register for the network and to sniff the Z-Wave packets;
- Wi-Fi and Ethernet: traditional network sniffing tools such as Wireshark or tcpdump may be used.

4.2.4. Unusual Activities and Battery-Depleting Attacks (N4)

Scenario: A wide variety of attacks may trigger additional processing in IoT devices. Therefore, ultimately, the presence of ongoing attacks and, in general, 'unusual activity' may be detected by monitoring battery consumption. An attacker may simply attempt to exhaust the system's energy or may carry out activities on or around the gateway for undetermined reasons. Such activities can signal various forms of malfunctions, anomalies

and even attacks and can constitute a simple means of monitoring the health and safety of the gateway.

Demonstration and validation methodology: The validation exploits the gathered traffic rates within the gateway and between the gateway and the outside world, recording normal traffic levels over extended periods. The validation will also measure energy consumption in various devices and the rates of battery depletion on a daily and, if possible, hourly basis. These metrics will be used to determine normal traffic rates and normal energy consumption, and they can be used to determine anomalies and attacks.

4.3. Software Attacks

While developing scenarios for software-related attacks, we observed that, as per the network properties, traditional risks remain persistent within the IoT landscape. In addition to traditional attacks, the following section depicts common scenarios and validation methodologies for compromised software attacks, command injection, mechanical exhaustion and sleep deprivation attacks.

4.3.1. Traditional Attacks (S1)

Scenario: Similar to traditional network attacks, the case of traditional software attacks implies the exploitation of traditional software flaws. Here, we find traditional attacks including virus infection exploits, worms and malicious script executions. In this context, we observe that the smart home installation, especially the gateway and the user devices that may be communicating with the gateway (e.g., a smartphone, laptop or smart TV), are built on traditional software systems that require periodic maintenance and security updates. In such a diverse technological ecosystem, it is easily conceivable that in a particular IoT installation, there may be out-of-date software with vulnerabilities that may be exploited by malicious actors. The attack showcases the execution of typical software exploits in the attempt to gain access to sensitive (IoT-specific) information. In this case, the attacker may exploit the vulnerabilities of services installed on the gateway or on one of the user devices (e.g., a laptop or smartphone) to obtain sensitive information.

Demonstration and validation methodology: Various attacks will be mounted against the gateway by emulating software flaws via network traffic that contains malware traces. The following tools will be used for this purpose:

- Packet capture (PCAP) files containing malware traces: malware traces shall be used from public sources (https://zeltser.com/malware-sample-sources/, accessed on 12 April 2022);
- `tcpreplay`, `tcprewrite`: these two tools are used to edit and replay the malware PCAP files against the gateway.

4.3.2. Compromised Software Attacks (S2)

Scenario: It is conceivable that a cybersecurity solution deployed in the smart home may not function properly, or it may get compromised by an attacker (as a consequence of software flaws and attack exploits). Consequently, the gateway will exhibit a different behaviour, such as stopping the sending anomaly alerts, or they may simply be stopped. Therefore, the user may be notified that the software is not functioning properly and that actions should be taken in order to ensure that attacks are detected. By exploiting software flaws, an attacker may succeed in changing one of the smart home software modules. The attacker may succeed in injecting new and malicious code that alters the solution's behaviour.

Demonstration and validation methodology: The attack demonstrates the awareness of the DRA regarding the malfunctioning or compromise of its own modules. This entails that each module detects the malfunctioning of the other modules it depends on. Obviously, the mechanisms implemented for detecting the changes in behaviour of its own software need to be as lightweight as possible and must build on simple decision algorithms and computations. Furthermore, given the overhead of more complex computations and the

limited hardware resources, only the most predictive behaviours shall be taken into account. In particular, the following expected operational behaviour shall be monitored:

- Network analysis: the periodic generation of PCAP files;
- Anomaly reporting: the periodic alerts issued;
- Configuration manager: the status of each OS process and the consumption of CPU and memory for each gateway OS process.

4.3.3. Command Injection (S3)

Scenario: Considering the software-oriented construction of an IoT system's inbuilt logic, it is reasonable to presume that, by exploiting software flaws, malicious software may be hosted by one of the IoT system's devices (e.g., a smartphone or gateway). Consequently, the malicious software may open legitimate communication channels for injecting commands into IoT devices. By exploiting software flaws, an attacker may succeed in running new software alongside a cybersecurity solution enabling sending of the forged commands to IoT devices sensors and communicating these data with an outside 'command and control' server.

Demonstration and validation methodology: The undertaken procedure includes additional software scripts hosted on the gateway and on an external device (e.g., a smartphone), from which the test, through legitimate commands, triggers relays and requests the status information on the sensor devices.

4.3.4. Mechanical Exhaustion (S4)

Scenario: Contrary to the mechanical exhaustion attack included in the list of physical attacks, software-oriented mechanical exhaustion presumes a malfunctioning software module or new software that sends ON/OFF commands to switching devices (e.g., relays) at a high rate. Considering that, inadvertently, a device has an upper limit in terms of mechanical switching, once that upper limit is exceeded, devices may malfunction, and they can be physically damaged.

Demonstration and validation methodology: The undertaken procedure shall include additional software scripts hosted on the gateway and on an external device (e.g., a smartphone), from which the tests, through legitimate commands, shall trigger relays at a higher rate than expected from normal operation.

4.3.5. Sleep Deprivation (S5)

Scenario: It is commonly known that in order to save energy, battery-powered IoT devices reduce their energy consumption by entering 'sleep mode'. Once an event is detected, the devices then resume their normal operation and forward the alert(s) to their associated controllers (e.g., the IoT gateway). However, the normal behaviour of IoT devices can be exploited by malicious actors in order to prevent devices from activating their energy-saving mode (i.e., the 'sleep mode'). An attacker can use software solutions to remotely launch this type of attack via legitimate commands. The commands may not necessarily need to cause the mechanical triggering of devices or a change of state. It would suffice to periodically request status information or to send a specifically forged packet that would prevent the device from entering its energy-saving mode.

Demonstration and validation methodology: The undertaken procedure shall include additional software scripts hosted on the gateway and on an external device (e.g., a smartphone). From these locations, the developed software shall repeatedly request the status of the devices, which in most cases should prevent the device from entering 'sleep mode'.

5. Technical Actions

In this section, we analyse the various attacks and derive a set of technical actions. We started out with a literature review and expert knowledge for aggregating an initial set of actions for each attack, which thereafter were verified and filtered by the manufacturers

through discussions. This process was repeated several times as the project implementation followed an iterative approach, where the technical capabilities of the gateways were improved over the course of time.

5.1. Physical Attack Actions

In the case of physical attacks, the supported actions are detailed in Table 4. Here, we observe that, in case of physical damage, the GHOST solution can mainly suggest some manual actions that the user should perform. Accordingly, the user should verify the physical integrity, and he or she should verify the battery level. However, other actions may also be taken, which may be automatable. To this end, 'sandboxing' may be used to isolate the device and its traffic from other devices. In this respect, 'one-way sandboxing' refers to application-level commands not routed to the IoT software's other modules (this is only applicable when there are also automated scripts available that logically link devices between them). A more restrictive scenario is 'two-way sandboxing', in which case the application-level commands (or notifications) will be blocked from being sent out to the specific device. Lastly, all traffic may be blocked to or from that particular device [37].

Next, in the case of device injection (i.e., a new device is detected), all actions may be automated. The device may simply be allowed to be added. However, the device may be sandboxed, where the meaning of the 'sandboxing' term is the one given above. Lastly, the device may be temporarily or permanently blocked.

In the last case of physical attacks, we find mechanical exhaustion. For this type of attack, the action may simply be permitted, or sandboxing can be used to limit its effects. Lastly, the device can simply be blocked (i.e., by dropping all traffic related to that particular device's flows).

Table 4. Technical actions in physical attacks.

Attack ID	Attack Name	Action	Action ID
P1	Damage	Verify physical integrity	T1
		Verify battery	T2
		One-way sandboxing	T3
		Two-way sandboxing	T4
		Block device (temp or perm)	T6, T7
P2	Device injection	Permit	T5
		One-way sandboxing	T3
		Two-way sandboxing	T4
		Block device (temp or perm)	T6, T7
P3	Mechanical exhaustion	Permit	T5
		One-way sandboxing	T3
		Two-way sandboxing	T4
		Block device (temp or perm)	T6, T7

5.2. Network Attack Actions

In the case of network-level attacks, the supported actions are detailed in Table 5. Here, we observe that all actions can be automated by the GHOST solution. In the case of traditional network attacks, packets may be dropped for a particular flow, or more dramatic measures may be taken in case of higher risk levels. To this end, all flows associated to a particular source may be dropped. The actions may be taken temporarily or permanently, depending on the attack's impact and its execution in time.

Next, in the case of device impersonation, one-way sandboxing is an effective measure to continue monitoring the flows associated to a device while ensuring that the application-level data do not reach other devices. Conversely, two-way sandboxing may be used to continue the monitoring of flows while ensuring that messages are not forwarded to the

possibly malicious device. Lastly, in this case as well, the device may be blocked, which means that all traffic to and from that particular device is dropped.

In terms of the side channel attack, since this type of attack is performed at the time of design, in order to analyse and infer the information that may be leaked from the GHOST solution, no actions are suggested.

Lastly, in the case of battery attacks, the same type of actions is defined as in the case of the traditional network attacks: drop the packets for a specific flow or for a specific source. Both actions are automatable.

Table 5. Technical actions in network attacks.

Attack ID	Attack Name	Action	Action ID
N1	Traditional	Drop packets for flow (temp or perm) Drop packets for source (temp or perm)	T8, T9 T10, T11
N2	Device impersonation	One-way sandboxing (temp or perm) Two-way sandboxing (temp or perm) Block device (temp or perm)	T3 T4 T6, T7
N3	Side-channel	(design time test only)	–
N4	Battery attacks	Drop packets for flow (temp or perm) Drop packets for source (temp or perm)	T8, T9 T10, T11

5.3. Software Attack Actions

In the case of network-level attacks, the supported actions are detailed in Table 6. Here, we observe that all actions can be automated by the GHOST solution. As expected, in the case of traditional malware attacks, packets associated to flows or to a particular source may be dropped temporarily or permanently. In the case of software compromise, the particular module or the complete GHOST solution may be restarted. In the same scenario, a module may be temporarily or permanently disabled, and update requests may be issued.

Next, in the case of command injection, mechanical exhaustion and sleep deprivation attacks, the GHOST solution may once again drop the packets associated to the particular flows or sources.

Table 6. Technical actions in software attacks.

Attack ID	Attack Name	Action	Action ID
S1	Traditional	Drop packets for flow (temp or perm) Drop packets for source (temp or perm)	T8, T9 T10, T11
S2	Software compromise	Restart module or GHOST Disable module (temp or perm) Send update request	T12,T13 T14, T15 T16
S3	Command injection	Drop packets for flow (temp or perm) Drop packets for source (temp or perm)	T8, T9 T10, T11
S4	Mechanical exhaustion	Drop packets for flow (temp or perm) Drop packets for source (temp or perm)	T8, T9 T10, T11
S5	Sleep deprivation	Drop packets for flow (temp or perm) Drop packets for source (temp or perm)	T8,T9 T10, T11

6. Decision Automation in Risk Assessment

The DRA at its core relies on three key innovation areas:

- Real-time risk assessment;

- Decision automation;
- Security usability.

These advances are fused together within the implementation of DRA [5], which relies on probabilistic traffic profiling, the intelligence on the threat and vulnerability likelihood and the associated risk's severity, which permits identifying network anomalies and coordinating the selection of the appropriate mitigation action.

The DRA incorporates a variety of analytic algorithms, called *analysers*, and each are responsible for the distinct features listed below:

- Behaviour analyser (BA): the main purpose of this analyser is to detect any deviation from the device's normal behaviour;
- Payload check (PC): a set of defined rules aiming to detect the presence of the user's sensitive data within the traffic flow;
- Block rules (BR): responsible for verifying the destination's maliciousness from personalised settings and common shared intelligence;
- Alert processor (AP): alert extraction analytics from external input for anomaly detection.

The resulting scores are incorporated together for predictive risk forecasting. Triggered by so-called *risk receptors*, a current probability of identified risk is estimated in conjunction with the user-defined risk level tolerance.

Table 7 contains the identified attacks and the associated analytic algorithms used for the risk evaluation.

Table 7. Attack and analyser mapping.

Attack ID	Apt Analyser	Rationale
P1	BA	Absence or change in behaviour of communication
P2	BA and AP	No behaviour profile present and alert propagation on non-registered device
P3	BA and AP	Absence or change in behaviour of communication and alert propagation on anomalous traffic
N1	AP	Alert propagation in threat detection
N2	BA and AP	Absence or change in behaviour of communication and alert propagation in anomalous traffic
N3	BA, PC and AP	Absence or change in behaviour of communication, presence of sensitive data and alert propagation in anomalous traffic
N4	BA and AP	Absence or change in behaviour of communication and alert propagation in anomalous traffic
S1	AP	Alert propagation in threat detection
S2	BA, PC and BR	Absence or change in behaviour of communication, presence of sensitive data and attempted communication with malicious destination
S3	BA and BR	Absence or change in behaviour of communication and attempted communication with malicious destination
S4	BA and AP	Absence or change in behaviour of communication and alert propagation in anomalous traffic
S5	BA and AP	Absence or change in behaviour of communication and alert propagation in anomalous traffic

We identified three different communication profiles allowing the end user to customise the automatic decision making when exceeding risk expectations as per the user preferences. The choice between three awareness modes is proposed through the configuration (CFG) interface. These modes are recapped in Table 8.

Finally, we derived several decision scenarios, which served as a basis for the conceptualisation of the decision tree, presented in Section 7:

- Missing communication (absence): Absence of the device's communication in relation to its normal behaviour;
- Whitelisting: New communication was neither blacklisted nor whitelisted before by the user or the GHOST solution itself;
- Data type (privacy): Private data leakage is tracked, and the user is informed of the violation of the policies defined;
- Frequency: A suspicious situation is detected in terms of the communication frequency, being too often or not frequent enough;

- Time (timing): The pattern of communication stays within the safe profile derived, but the actual timing is suspicious;
- Blacklisting: Known illegitimate communication is taking place, but it is generating activity on the internal network despite being blocked from further external propagation.

The sparse automation matrix is presented in Table 9, mapping the above defined awareness modes, decision scenarios and automation feasibility dictated by risk acceptance agreement.

Table 8. Awareness preferences.

Name	Description	Informed on Any Decision	Allow Risk-Controlled Automation
Raise Awareness	Stay informed of any decisions that GHOST made by displaying a corresponding notification. The GHOST system will automatically block any suspicious communication as soon as maximum risk level is exceeded.	Yes	Yes
Enforced Awareness	Stay informed of any decisions that GHOST made by displaying a corresponding notification. GHOST will not perform any automatic decisions when the maximum risk level is exceeded. One will be constantly prompted to review suggested actions and make decisions by one's self.	Yes	No
Problem Awareness	Stay informed of decisions that GHOST made only when exceeding maximum risk level by displaying a corresponding notification. GHOST will automatically block any suspicious communication as soon as maximum risk level is exceeded.	No	Yes

Table 9. Sparse decision automation matrix.

	Absence	Whitelisting	Privacy	Frequency	Timing	Blacklisting
Raise	-	Automatable	Automatable	Automatable	Automatable	Automatable
Enforced	-	-	-	-	-	Automatable
Problem	-	Automatable	Automatable	Automatable	Automatable	Automatable

7. Decision Tree Conceptualisation

The scenarios from the initial scenario definition were further transformed into a decision tree, focusing on the essential interaction with the end user and aiming at the enlightening comprehension and in-depth involvement of their feedback through user studies on the end users' mental models. The overall view of the decision tree implemented for the first prototype is depicted in Figure 2. The following section sets the basis for understanding the main decision branches, the DRA's configuration set-up scope and how the DRA's monitoring is automated.

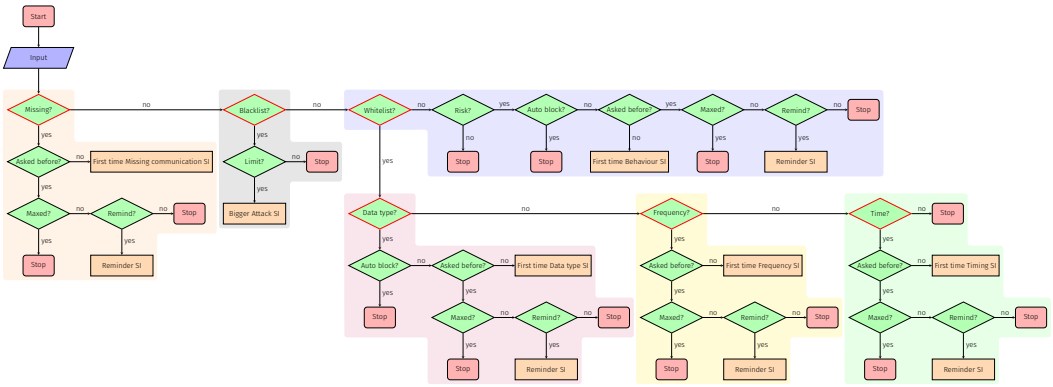

Figure 2. Decision tree conceptualisation with coloured highlighting of the main branches.

7.1. Decision Branches

As shown in Figure 2, the distinct decision branches have been grouped by colour within the tree and are described in the section hereafter.

7.1.1. Missing Communication

The first group of interfaces covers the case where the DRA detects the absence of device communication in relation to its normal behaviour. The close-up of the decision tree is depicted in Figure 3. This module includes a standard security intervention (SI) interface with a possible reminder of the pending user's decision, allowing him or her to ignore the situation until an automated decision is made to treat this case as safe.

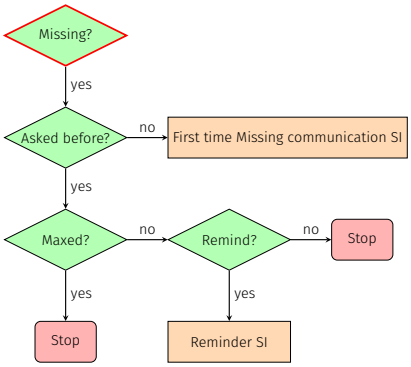

Figure 3. Missing communication.

7.1.2. Whitelisting

The second group of the interfaces refers to the case where a new communication was neither blacklisted nor whitelisted before by the user or the GHOST solution itself. As demonstrated in Figure 4, the DRA will make a risk-based decision based on the likelihood of the maliciousness of the destination party, the actual profile of the communicating device and also user risk acceptability. This user interface is also composed of an initial Security Intervention (SI) interface and a reminder of a pending decision in the case where automatic decision cannot be made (according to the user preference).

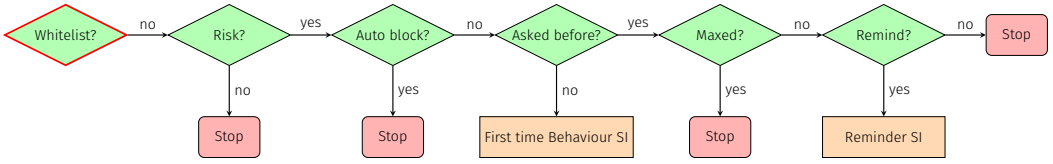

Figure 4. Whitelisting.

7.1.3. Data Type

The third type of SI interface comes into play in light of the privacy controls. Depending on the end user's configuration, certain private data leakage will be tracked, and the user will be informed of the violation of the policies defined. This interface is also composed of a first-time SI and a consecutive reminder if necessary.

7.1.4. Frequency

The fourth type of user interface is focused on making decisions when a suspicious situation is detected in terms of the communication frequency, occurring either too often or not frequently enough. Once again, this interface is composed of first-time SI and the reminder for the pending decision. Once the limit is passed over, the notification will be completely discarded.

7.1.5. Time

The fifth group of SI interfaces covers the case where the pattern of communication stays within the safe profile derived, but the actual timing is suspicious. In this case, a notification will be sent to the user to confirm if this timing is appropriate as shown in Figures 5–7.

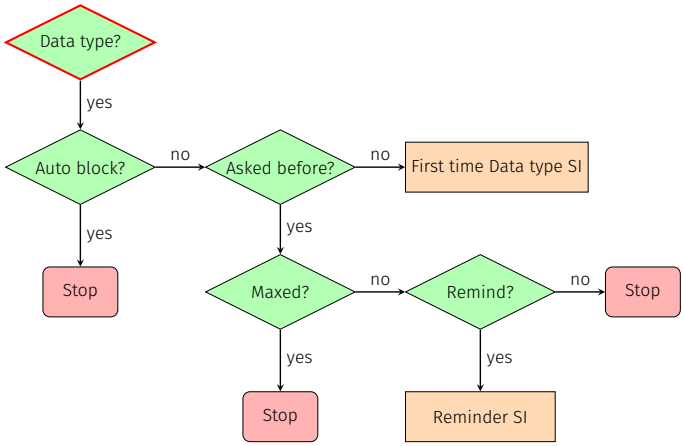

Figure 5. Data type.

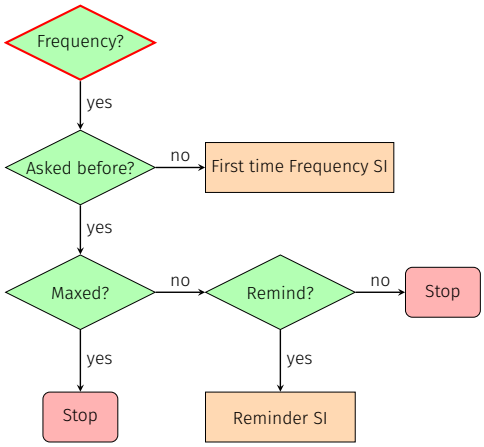

Figure 6. Frequency.

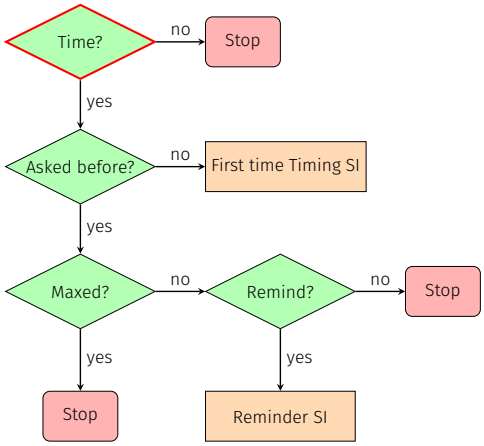

Figure 7. Time.

7.1.6. Blacklisting

As shown in Figure 8, the last group of interfaces is preserved for the situations with mitigation propositions in cases where it is known that communication is illegitimate, but it is generating activity on the internal network despite being blocked from further external propagation. In such a case, we suspect an IoT device to be part of the bigger attack such as a DDoS and possibly affecting the performance of the actual device.

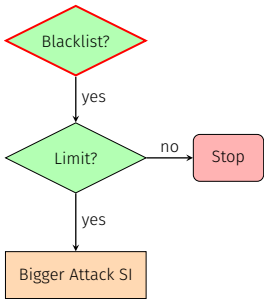

Figure 8. Blacklisting.

7.2. Configuring the DRA

The main focus of this type of interface is given to the effortless and usable design of the configuration set-up process and the further settings review and fine-tuning of the applied configuration policies, called configuration (CFG). This design and implementation was performed in four iteration cycles, with each being stipulated by the feedback received from the end users and fed back to the decision tree conceptualisation:

1. The development approach was based on the requirements derived from literature research and the results from the first set of user studies. Furthermore, the categorisation of the initial set of navigation pages was developed and is outlined in Table 10.
2. For the second iteration, the Configuration (CFG) interfaces were improved in terms of their usability with a mobile device form factor, and a new menu was developed for easier access to the different CFG sections. Furthermore, the colour theme was updated to match the official project theme.
3. The third iteration of the configuration was refined based on the results from the second set of user studies and the derived requirements. The updated specifications for the categorisation of the initial set of navigation pages developed are shown in Table 11.
4. The fourth cycle was mostly based on the perception of risk, requiring clarification of the associated impact. The final look of the CFG interfaces is depicted in Figures 9–11.

Table 10. Configuration interface: initial set-up.

Category	Specification
Welcome	Initial welcome screen
User registration	Initial user registration and authentication set-up
Dedicated device registration	User's interfacing device registration
Smart home environment	Configuration of the smart home, aiming to provide settings for: • Adding and removing IoT devices • Configuration of 'unknown' devices • Naming and custom identification of IoT devices
Mode selection	Three configuration modes are envisioned to target different user profiles: • Manual: based on predefined settings • Assistant: step-by-step configuration • Delegation: configuration by trusted third party
Step-by-step configuration	Detailed configuration of the GHOST main features: • Blocking rules: customisation for blacklisting or whitelisting the communication destination parties • Acceptable risk level: definition of the permitted risk levels for security and privacy settings, defining a threshold for DRA automated decisions • Privacy monitor: selection of private data categories for tracking • Awareness requirements: configuration of the desired intervention and involvement level in decision making

Table 11. Configuration interface: updates to the initial set-up.

Category	Specification
Mode selection	The three configuration modes are updated to differentiate between different target user profiles: • Manual: based on predefined settings • Delegation: configuration by trusted third party • Advanced: for expert level configurations (e.g., whitelist or blacklist)
Step-by-step configuration	Detailed configuration of the GHOST main features: • Security preferences: definition of the accepted risk levels for security after which all communications will be blocked • Privacy preferences: selection of private data categories not to be transferred to the Internet • Notification preferences: configuration of the corresponding preference for security and privacy notifications

7.3. Monitoring Automated DRA

The SI component aims to develop a type of user interfaces for user-friendly visualisation of risk tracking, and the risk evaluation results and will enable the end user to make informative decisions. The overall technological selection and implementation refinement process is closely aligned with the CFG interfaces. Five iteration cycles were implemented, each being stipulated by the feedback received from the end users and fed back to the decision tree conceptualisation:

1. The first iteration of the security intervention interface was developed based on the requirements from the initial literature review. Furthermore, the initial listing of possible interactions with the GHOST solution was created. This outline is summarised in Table 12.
2. To provide a more fluent and unified experience, the SI notifications and related front-end were included in the same Angular web application package as the CFG interface.

3. The second prototype was developed based on the input from the user trials, particularly by attempting to provide a more understandable and actionable input for the user's decisions.
4. The existing SI was further fine-tuned in preparation for the third trials, where attack simulations would be performed. For this purpose, not only were additional menu items were added, but the generic system flow was also amended to reassure the end user and provide additional information on the ongoing evaluation and feedback gathering.
5. The final iteration was mostly concerned with the proper naming of the awareness preferences, which were previously outlined in Table 8. The final look of the SI interfaces is depicted in Figures 12–14.

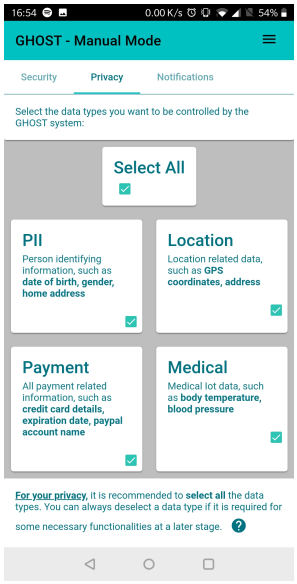

Figure 9. Privacy preferences.

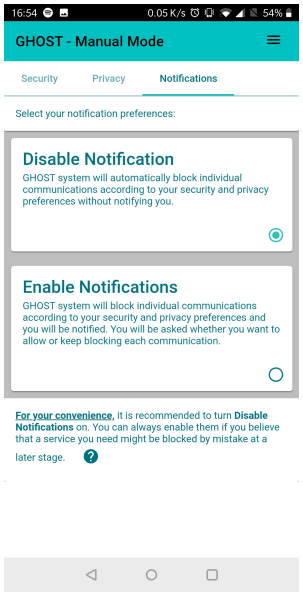

Figure 10. Notification preferences.

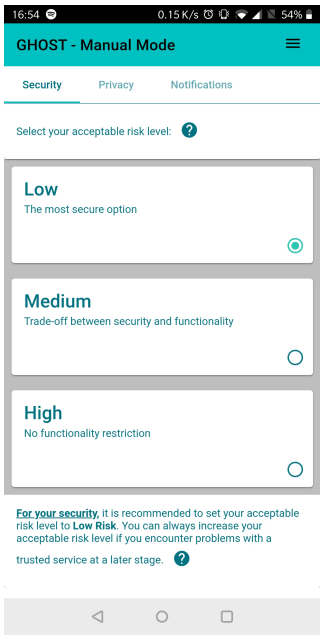

Figure 11. Security preferences.

Table 12. Security intervention: interaction identification.

Category	Specification
Mismatching device behaviour	• Absence of or decrease in communication • Increase of or frequency change in communication • Extra 'steps' in communication
Communication with blocked src or dst	Attempt to initiate communication with blocked party
'New' src or dst (unknown)	• Fetching contextual information on the new party (e.g., 'whois' information) • Consecutive configuration update
Device parameter anomalies	An overview of the device profile and its activity (e.g., battery power)
Payload-related	• Security-related: clear text password of credit card • Privacy-related: PII data (date of birth, home address) • Secured transmission security check • Masking of data in encrypted communication channels
Mitigation action	Hypothesis presentation with possible suggestion for mitigation recommendation
Current status	Display of the risk level's current status in relation to defined accepted level for security and privacy risks
Predicted status	A risk level estimation representation after the evaluation of the current communication-associated risk
Impacts (text)	Possible impact score value
Recommendations	Mitigation action to support the decrease of the raised risk level
History of risk assessments	Interface link to historical data visualisation
History on suspicious activity	Interface link to historical data visualisation

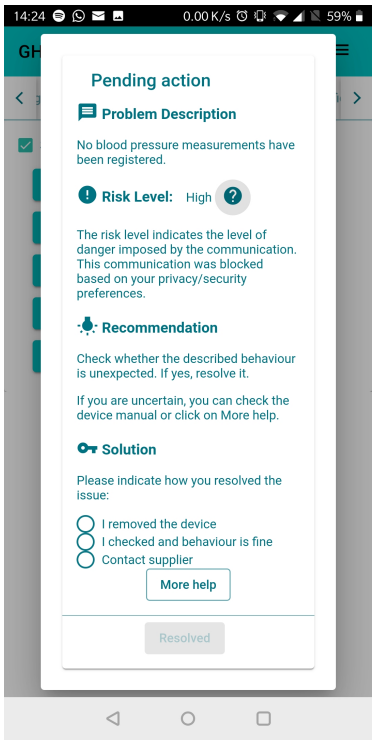

Figure 12. Pending action.

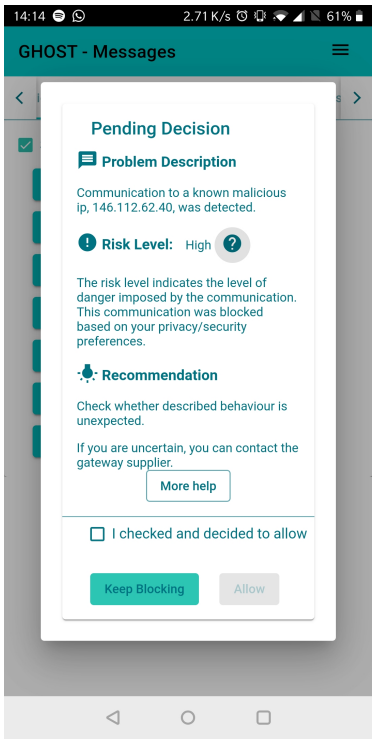

Figure 13. Pending decision.

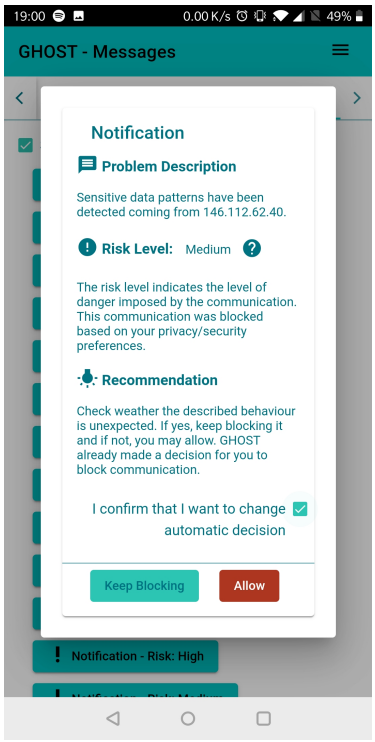

Figure 14. Notification of automated decision.

8. Discussion and Future Work

This section presents a summary of our findings through dedicated analysis of each research question:

RQ1 Limitations on automation: We started our research with the identification of possible technical actions to mitigate the exposure to smart home threats. As outlined in Section 5, for each category of the attacks, we derived a list of technical actions further linked to the automation feasibility, as presented in Table 2. As can be observed, only actions of a physical nature (such as physical verification of the device integrity or battery state) were not possible to automate. However, we were able to address this through inclusion of the actions in mitigation advisory, enabling guidance to the user.

RQ2 Usable actions translation: Guided by the user studies and continuous feedback collection, we were able to derive a short set of usable actions to be presented as part of the final UI. The process we followed transformed the initial interfaces significantly with the minimum information on the detailed and fine-tuned textual descriptions to have maximum user engagement. Limiting the number of usable actions had a positive effect on the usability aspects of the final interfaces, which was showcased during real-life pilot deployments with an average System Usability Scale (SUS) score [38] that increased throughout the project's lifetime.

RQ3 Perception's linkability to engagement: The differences and impact of personal risk perception were a key challenge that we addressed in this research. As pointed out by Gerber et al. [17], 'people tend to base their decisions on perceived risk instead of actual risk', and this human aspect complicates the translation of technical risk information into a format that would engage the user in his or her digital security and privacy exposure. As a result, for the risks emanating from the users' IoT assets, which are often not directly perceivable, the users are unaware of the consequences. While we derived a set of usable actions, it neither directly engages the user nor provides any preferences for how much a user wants to be in control or, for that matter, be informed. For this purpose, we proposed and developed a decision tree concept and three types of awareness preferences (outlined in Sections 6 and 7, respectively). While the provided solution was refined through four iterations and proved to be successful

during the project deployments, the inadvertent limitation by the published works on the user studies should be noted for future research [16–18], as they were conducted mainly in Germany and their samples were likely to be biased due to the usage of an online recruitment panel. With our proposed solution, we provide the means to address the security and privacy perception of each individual user while preserving a high level of security through balancing the automation and security preferences.

As another future work, a greater emphasis on privacy attacks and compliance is envisioned. Our ongoing efforts are focused on evaluating regulatory and standardisation bodies' efforts in deploying privacy-preserving techniques in the IoT environment in general. Future work seeks to refine the currently proposed framework through integration of the key up-to-date regulations and standards.

As for a limitation of this work, the present research was confronted with data generation challenges and validation obstacles. Within the GHOST project, it was initially planned to generate IoT data and simulate attacks from real-life smart homes. However, due to ethical and privacy restrictions, the scope was adjusted to testbed environments. To that end, our developed user-centric decision tree was built over a limited selection of testbed attacks that can be elevated further with a more thorough list of known and unknown attacks. On that same note, more granular expert validation can be reported over real-life anomalies, which will naturally refine the decision-making process discussed in Section 7.

9. Conclusions

In this work, we presented a methodological framework applied in the design and implementation of usable interfaces for the DRA-enabled smart home environment. Guided by the definition of the threat vectors, we identified a set of technical actions suitable for threat prevention. Those were further translated into usable actions, meaning understandable digital decisions which an end user of the smart home with different technological knowledge can make. This flow from the threat vector to technical actions is then translated into usable actions and sets the basis for the decision tree concept, where each branch is provided with a reasoning and execution flow, which further translates the usable actions in an iterative manner into two types of UIs: CFG and SI. While most of the technical actions were translated into automated mitigation actions, through the awareness preferences concept, a mitigation advisory allowed the handling of those remaining non-automated technical actions.

Author Contributions: Conceptualisation, A.C., I.-C.S., B.G. and N.A.N.; methodology, A.C. and N.A.N.; software, A.C.; validation, A.C.; writing—original draft preparation, A.C.; writing—review and editing, A.C., M.B. and N.A.N.; visualisation, N.A.N. and A.C.; supervision, N.A.N. and B.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded and supported by the European Union's Horizon 2020 Research and Innovation Programme through the AVENUE project (https://h2020-avenue.eu/, accessed on 12 April 2022) under grant agreement No. 769033, nIoVe project (https://www.niove.eu/, accessed on 12 April 2022) under grant agreement No. 833742 and SHOW project (https://show-project.eu/, accessed on 12 April 2022) under grant agreement No. 875530.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

AP	Alert processor
BA	Behaviour analyser
BR	Block rules
CFG	Configuration
DDoS	Distributed denial-of-service
DoS	Denial-of-service
DRA	Dynamic risk assessment
IoT	Internet of Things
IR	Information retrieval
ISys	Information system
PC	Payload check
PCAP	Packet capture
RA	Risk assessment
SI	Security intervention
SUS	System Usability Scale
UI	User interface
UX	User experience

References

1. Bansal, M.; Chana, I.; Clarke, S. A Survey on IoT Big Data. *ACM Comput. Surv.* **2021**, *53*, 131. [CrossRef]
2. Almusaylim, Z.A.; Zaman, N. A review on smart home present state and challenges: Linked to context-awareness internet of things (IoT). *Wirel. Netw.* **2019**, *25*, 3193–3204. [CrossRef]
3. Assal, H.; Chiasson, S. "Think secure from the beginning": A survey with software developers. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; pp. 1–13. [CrossRef]
4. Jacobsson, A.; Boldt, M.; Carlsson, B. A risk analysis of a smart home automation system. *Future Gener. Comput. Syst.* **2016**, *56*, 719–733. [CrossRef]
5. Collen, A.; Nijdam, N.A. Can I Sleep Safely in My Smarthome? A Novel Framework on Automating Dynamic Risk Assessment in IoT Environments. *Electronics* **2022**, *11*, 1123. [CrossRef]
6. Haim, B.; Menahem, E.; Wolfsthal, Y.; Meenan, C. Visualizing insider threats: An effective interface for security analytics. In Proceedings of the 22nd International Conference on Intelligent User Interfaces Companion, Limassol, Cyprus, 13–16 March 2017; pp. 39–42. [CrossRef]
7. Realpe, P.C.; Collazos, C.A.; Hurtado, J.; Granollers, A. Towards an integration of usability and security for user authentication. In Proceedings of the XVI International Conference on Human Computer Interaction, Vilanova i la Geltru, Spain, 7–9 September 2015; p. 43. [CrossRef]
8. Preibusch, S. Privacy behaviors after Snowden. *Commun. ACM* **2015**, *58*, 48–55. [CrossRef]
9. Dhillon, G.; Oliveira, T.; Susarapu, S.; Caldeira, M. Deciding between information security and usability: Developing value based objectives. *Comput. Hum. Behav.* **2016**, *61*, 656–666. [CrossRef]
10. Andriotis, P.; Oikonomou, G.; Mylonas, A.; Tryfonas, T. A study on usability and security features of the Android pattern lock screen. *Inf. Comput. Secur.* **2016**, *24*, 53–72. [CrossRef]
11. Lee, J.H.; Kim, H. Security and Privacy Challenges in the Internet of Things [Security and Privacy Matters]. *IEEE Consum. Electron. Mag.* **2017**, *6*, 134–136. [CrossRef]
12. Alur, R.; Berger, E.; Drobnis, A.W.; Fix, L.; Fu, K.; Hager, G.D.; Lopresti, D.; Nahrstedt, K.; Mynatt, E.; Patel, S.; et al. Systems Computing Challenges in the Internet of Things. *arXiv* **2016**, arXiv:1604.02980.
13. Nurse, J.R.; Atamli, A.; Martin, A. Towards a usable framework for modelling security and privacy risks in the smart home. In Proceedings of the 4th International Conference on Human Aspects of Information Security, Privacy, and Trust, Toronto, ON, Canada, 17–22 July 2016; Volume 9750, pp. 255–267.
14. Dutta, S.; Madnick, S.; Joyce, G. SecureUse: Balancing security and usability within system design. In Proceedings of the 18th International Conference on Human-Computer Interaction, Toronto, ON, Canada, 17–22 July 2016; Communications in Computer and Information Science; Volume 617, pp. 471–475.
15. Augusto-Gonzalez, J.; Collen, A.; Evangelatos, S.; Anagnostopoulos, M.; Spathoulas, G.; Giannoutakis, K.M.; Votis, K.; Tzovaras, D.; Genge, B.; Gelenbe, E.; et al. From Internet of Threats to Internet of Things: A Cyber Security Architecture for Smart Homes. In Proceedings of the 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Limassol, Cyprus, 11–13 September 2019; pp. 1–6. [CrossRef]
16. Gerber, N.; Reinheimer, B.; Volkamer, M. Home Sweet Home? Investigating Users' Awareness of Smart Home Privacy Threats. In Proceedings of an Interactive Workshop on the Human Aspects of Smarthome Security and Privacy (WSSP), Baltimore, MD, USA, 12 August 2018.
17. Gerber, N.; Reinheimer, B.; Volkamer, M. Investigating People's Privacy Risk Perception. *Proc. Priv. Enhancing Technol.* **2019**, *2019*, 267–288. [CrossRef]

18. Duezguen, R.; Mayer, P.; Berens, B.; Beckmann, C.; Aldag, L.; Mossano, M.; Volkamer, M.; Strufe, T. How to Increase Smart Home Security and Privacy Risk Perception. In Proceedings of the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 20–22 October 2021; pp. 997–1004. [CrossRef]
19. Yee, K.P. Aligning security and usability. *IEEE Secur. Priv.* **2004**, *2*, 48–55. [CrossRef]
20. Caputo, D.D.; Pfleeger, S.L.; Sasse, M.A.; Ammann, P.; Offutt, J.; Deng, L. Barriers to Usable Security? Three Organizational Case Studies. *IEEE Secur. Priv.* **2016**, *14*, 22–32. [CrossRef]
21. Balfanz, D.; Durfee, G.; Smetters, D.; Grinter, R. In search of usable security: Five lessons from the field. *IEEE Secur. Priv. Mag.* **2004**, *2*, 19–24. [CrossRef]
22. Atzeni, A.; Faily, S.; Galloni, R. Usable Security. In *Encyclopedia of Information Science and Technology*, 4th ed.; Khosrow-Pour, D.B.A., Ed.; IGI Global: Hershey, PA, USA, 2018; Chapter 433, pp. 5004–5013. [CrossRef]
23. Barbosa, N.M.; Zhang, Z.; Wang, Y. Do Privacy and Security Matter to Everyone? Quantifying and Clustering User-Centric Considerations About Smart Home Device Adoption. In Proceedings of the Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020), Online, 7–11 August 2020; pp. 417–435.
24. Emami-Naeini, P.; Agarwal, Y.; Faith Cranor, L.; Hibshi, H. Ask the experts: What should be on an IoT privacy and security label? In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2020; pp. 447–464. [CrossRef]
25. Tabassum, M.; Carolina, N.; Kosinski, T.; Clara, S. "I don' t own the data": End User Perceptions of Smart Home Device Data Practices and Risks. In Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security, Santa Clara, CA, USA, 11–13 August 2019; pp. 435–450.
26. 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]
27. Vervier, P.A.; Shen, Y. Before toasters rise up: A view into the emerging IoT threat landscape. In Proceedings of the 21st International Symposium on Research in Attacks, Intrusions, and Defenses, Heraklion, Greece, 10–12 September 2018; pp. 556–576.
28. Haney, J.M.; Furman, S.M.; Acar, Y. Smart home security and privacy mitigations: Consumer perceptions, practices, and challenges. In Proceedings of the 2nd International Conference on Human-Computer Interaction, Copenhagen, Denmark, 19–24 July 2020; pp. 393–411.
29. Dunphy, P.; Vines, J.; Coles-Kemp, L.; Clarke, R.; Vlachokyriakos, V.; Wright, P.; McCarthy, J.; Olivier, P. Understanding the Experience-Centeredness of Privacy and Security Technologies. In Proceedings of the 2014 workshop on New Security Paradigms Workshop—NSPW '14, Victoria, BC, Canada, 15–18 September 2014; ACM Press: New York, NY, USA, 2014; pp. 83–94. [CrossRef]
30. Collard, H.; Briggs, J. Creative Toolkits for TIPS. In Proceedings of the ESORICS 2020: European Symposium on Research in Computer Security, Guildford, UK, 17–18 September 2020; Boureanu, I., Druagan, C.C., Manulis, M., Giannetsos, T., Dadoyan, C., Gouvas, P., Hallman, R.A., Li, S., Chang, V., Pallas, F., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 39–55.
31. Victora, S. IoT Guard: Usable Transparency and Control over Smart Home IoT Devices. Ph.D. Thesis, Institut für Information Systems Engineering, Montréal, QC, Canada, 2020. [CrossRef]
32. Bada, M.; Sasse, A.M.; Nurse, J.R.C. Cyber Security Awareness Campaigns: Why do they fail to change behaviour? *arXiv* **2019**, arXiv:1901.02672.
33. Chalhoub, G.; Flechais, I.; Nthala, N.; Abu-Salma, R.; Tom, E. Factoring user experience into the security and privacy design of smart home devices: A case study. In Proceedings of the CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020; pp. 1–9. [CrossRef]
34. Feth, D.; Maier, A.; Polst, S. A user-centered model for usable security and privacy. In Proceedings of the 5th International Conference on Human Aspects of Information Security, Privacy, and Trust, Vancouver, BC, Canada, 9–14 July 2017; pp. 74–89.
35. Grobler, M.; Gaire, R.; Nepal, S. User, Usage and Usability: Redefining Human Centric Cyber Security. *Front. Big Data* **2021**, *4*, 583723. [CrossRef] [PubMed]
36. 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]
37. Li, Y.; McCune, J.; Baker, B.; Newsome, J.; Drewry, W.; Perrig, A. Minibox: A two-way sandbox for x86 native code. In Proceedings of the 2014 USENIX Annual Technical Conference, USENIX ATC 2014, Philadelphia, PA, USA, 19–20 June 2014; USENIX Association: Philadelphia, PA, USA, 2014; pp. 409–420.
38. Lewis, J.R. The System Usability Scale: Past, Present, and Future. *Int. J. Hum.-Comput. Interact.* **2018**, *34*, 577–590. [CrossRef]