



SecureTAP: A Conversational Agent for Secure and Privacy-Aware Smart Home Automations

Arianna Di Serio
HIIS Laboratory
CNR-ISTI
Pisa, Italy
arianna.diserio@isti.cnr.it

Simone Gallo
HIIS Laboratory
CNR-ISTI
Pisa, Italy
simone.gallo@isti.cnr.it

Fabio Paternò
HIIS Laboratory
CNR-ISTI
Pisa, Italy
fabio.paterno@isti.cnr.it

Abstract

The rapid growth of Internet of Things (IoT) devices in smart homes presents significant opportunities for enhanced convenience and automation but also introduces notable challenges, particularly in terms of security, privacy, and understandability for non-expert users. This paper presents SecureTAP, a conversational agent designed to assist users in creating and modifying trigger-action programming (TAP) automations in smart home environments. The system leverages GPT-4o's capabilities to identify potential security vulnerabilities and privacy concerns in automation rules, offering users proactive mitigation strategies. Through an iterative process of prompt engineering, we developed a system that analyses automations, identifies risks, and suggests safer configurations to safeguard user privacy and security in IoT smart spaces. A user study with 15 participants evaluated the effectiveness of SecureTAP in addressing security and privacy issues, as well as user trust in the assistant's recommendations. Results indicated that SecureTAP effectively simplified the automation process while raising users' awareness of potential security and privacy concerns.

CCS Concepts

• **Human-centered computing**; • **Human computer interaction (HCI)**; • **Interaction paradigms**; • **Natural language interfaces**;

Keywords

Internet of Things, Conversational Interfaces, Trigger-Action Automations, Security and Privacy

ACM Reference Format:

Arianna Di Serio, Simone Gallo, and Fabio Paternò. 2025. SecureTAP: A Conversational Agent for Secure and Privacy-Aware Smart Home Automations. In *CHIItaly 2025: 16th Biannual Conference of the Italian SIGCHI Chapter (CHIItaly 2025)*, October 06–10, 2025, Salerno, Italy. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3750069.3750318>

1 Introduction

The increasing adoption of Internet of Things (IoT) devices in smart homes has transformed daily life, offering enhanced convenience, automation, and energy efficiency. According to Statista, the number of connected IoT devices worldwide is projected to reach 30.9

billion by 2025, up from an estimated 13.8 billion in 2021¹. However, this rapid proliferation also brings significant challenges, including security vulnerabilities, privacy concerns, and usability barriers, especially for non-expert users [1], [3], [7], [18], [23].

Research highlights that many users struggle with configuring smart home automations due to complex interfaces and limited programming knowledge. Indeed, traditional platforms often require manual rule creation, which can be unintuitive and error-prone, leading to frustration and ineffective automations [9]. Moreover, studies indicate that users frequently encounter negative experiences with smart home devices, such as automation failures, lack of transparency in system behaviour, and security concerns [4], [11]. These issues are exacerbated by the growing interconnectivity of devices, increasing the risk of unauthorized access or data leaks. Security and privacy risks are particularly critical in smart home environments. Automation rules, if not properly configured, can lead to unintended security breaches or privacy violations, such as exposing sensitive data or enabling unauthorized control over home devices [3]. Despite the existence of various security frameworks, many users lack the expertise to assess potential risks associated with their automation rules, highlighting the need for more accessible and intelligent tools.

To address these challenges, we introduce SecureTAP, a conversational agent designed to assist users in creating, validating, and securing smart home automations through natural language interaction. This new system considers a previous conversational agent [10], which demonstrated feasibility and usability in supporting users with creating and modifying trigger-action automations through natural language. The new assistant not only simplifies the process of writing trigger-action programming (TAP) rules but also proactively identifies potential security and privacy issues. The system evaluates automation rules against a predefined set of six risk categories (described in Section 3.1), offering contextual explanations and suggesting safer alternatives where needed.

To assess the usefulness of SecureTAP, we conducted a user study involving 15 participants to address the following research questions:

- How effective and useful is SecureTAP in helping users identify and prevent security and privacy vulnerabilities in smart home automation rules?
- To what extent do users trust the assistant's suggestions and mitigation strategies regarding automation safety?

The paper is organized as follows. After the introduction, we discuss related work concerning conversation agents that help users create



This work is licensed under a Creative Commons Attribution 4.0 International License. *CHIItaly 2025, Salerno, Italy*

© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2102-1/2025/10
<https://doi.org/10.1145/3750069.3750318>

¹<https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/>

automation for their smart environments, eventually with the aim of considering security and privacy. Next, we illustrate the design and implementation of the proposed solution. We also report on a user test, its results and the associated data analysis. Section 5 discusses the lessons learnt from the study. Lastly, we present some concluding remarks and indications for future work.

2 Related work

Several studies have investigated automation and virtual assistants in smart home environments, primarily focusing on usability and interaction paradigms. For instance, Li et al. [15] and Corno et al. [7] developed conversational AI systems capable of translating natural language commands into automation rules, enabling a no-code experience. Similarly, Kadali et al. [13] and Muslih et al. [17] employed chat-based interfaces for controlling IoT devices. While these systems streamline user interaction, they largely neglect the security implications of automation logic.

In contrast, security-aware approaches remain limited. A notable example is the work by Breve et al. [3], which addresses security concerns by using Hybrid Prompt Learning to generate natural language justifications for potential risks embedded in IF-THEN rules. Their evaluation showed that AI-generated explanations improved user understanding by up to 27% over traditional methods. However, their system focuses on post-hoc explanations rather than real-time prevention or mitigation.

Other contributions combine home automation with security features, though without analysing the automation logic itself. For instance, one study [20] integrates chatbot-based control via Facebook Messenger with a facial recognition door lock and emergency alerts for events like gas leaks or fires. While effective in handling physical threats, it does not assess, or address security risks introduced by user-defined automation rules. Similarly, another work [19] examines conflicts arising from concurrent user commands in shared environments but does not consider security threats related to rule configurations.

Recent research has also explored the role of large language models (LLMs) in smart home contexts. A study [12] evaluated GPT-4’s ability to infer routines, detect anomalies, and generalise behavioural patterns from sensor data using one-shot prompting. While the model could identify certain patterns, it struggled to generalise reliably over time. Notably, the authors emphasised the privacy risks associated with using LLMs to analyse sensitive household data, as such capabilities could unintentionally expose personal habits and behaviours.

In summary, while previous chatbot-based systems [7], [13], [15], [17] have primarily emphasised the system usability and efficiency, and others [3], [12], [20] have tackled isolated security or privacy issues, none have integrated these concerns holistically into the automation creation process. SecureTAP fills this gap by embedding real-time risk assessment and mitigation into natural language interactions, allowing users to create, validate, and secure automation rules proactively. Unlike post-hoc approaches such as [3], SecureTAP offers immediate feedback and guidance before a rule is deployed.

As smart homes become increasingly autonomous, the likelihood of unintentionally introducing unsafe or privacy-compromising

behaviours also rises. Embedding preventive security measures at the point of automation design is therefore a crucial step toward building trustworthy and resilient smart environments.

3 The Proposed Solution

To address security challenges in smart home automation, we developed SecureTAP, a conversational agent that assists users in creating, validating, and securing automation rules through natural language interaction. The prototype considers previous studies [10] that introduced a conversational agent capable of translating user intents into structured TAP automations through a chat-based interface. In this new solution, we focus on introducing a real-time reasoning support between rule generation and deployment. Specifically, each automation is analysed by a dedicated GPT-4o instance using a custom prompt (Section 3.1) to identify potential security or privacy issues (proposing safer alternative) before the rule is finalised. The vulnerabilities we aimed to identify include automation manipulation, network threats, access and authentication issues, TAP errors, user misconfigurations, and unsafe device interactions.

To demonstrate the practical value of the proposed conversational assistant, we describe representative use cases based on common smart home configurations tested with the novel proposed assistant. Each example highlights how the system identifies potential security, or privacy risks and provides mitigation strategies.

Voice-Controlled Door Unlocking via Smart Speaker. A user configures a rule that unlocks the front door when a specific voice command is issued to their smartphone’s voice assistant, aiming to enable hands-free access while approaching home. SecureTAP flags a serious security risk: if the phone is lost, stolen, or left unattended, unauthorised individuals could exploit voice control to gain entry. SecureTAP identifies a significant security flaw and recommends enabling user-specific voice profiles combined with proximity verification (e.g., via Wi-Fi connection).

Ventilation Triggered by Indoor Temperature. The user defines a rule to open windows when the temperature exceeds 30°C, to cool the indoor environment during hot days. SecureTAP flags that opening windows without verifying if someone is home may increase exposure to burglary or allow unauthorised access and suggests incorporating presence detection or motion sensors to ensure that windows will be opened only when occupants are present or the surrounding area is secure.

Night-Time Power-Off of Surveillance Devices for Privacy. To preserve personal privacy during evening hours, a user configures a rule to automatically disable all indoor cameras and microphones after 10 PM. SecureTAP flags a vulnerability: disabling surveillance uniformly and without context could provide a time window for intruders to go undetected. It suggests a “context-aware” shutdown based on the presence and known safety of all household members and maintaining recording fallback modes that trigger only under unusual or risky patterns (e.g., front door or windows open late at night, detected distress sounds).

3.1 The Prompt

To accomplish our goal, we designed a prompt to evaluate a new automation rule during the automation creation process for a given

configuration of the user’s smart home. The prompt guides the system in detecting and explaining common classes of security and privacy vulnerabilities that may emerge from smart home automations. Specifically, it evaluates each rule against a predefined set of six issue categories:

- Automation Manipulation, when an attacker manipulates automations for their own benefit.
- Network Threats (e.g., MITM, Network sniffing, DoS), when communication between devices is exposed due to a lack of encryption or weak protocols.
- Access or Authentication Issues (e.g., unauthorised access, over-privilege), when devices or rules grant permission too broadly, or lack checks for user identity.
- TAP Errors, such as conflicts or unintended interactions between rules, which may cause devices to behave unpredictably.
- Unintentional User Misconfigurations, errors made by users when configuring or managing devices and automations, leading to vulnerabilities or malfunctions.
- Unsafe Device Interactions, when two or more devices interact in a way that creates unforeseen consequences.

For each category we provided a detailed description of the issue, examples of sub-categories, possible causes and consequences, detection conditions, common patterns, mitigation strategies, severity (high, medium-high, low), and concrete examples of automation rules that may lead to these problems. For example, for the category “Network threats” the sub-categories provided were DoS, man-in-the-middle, and network sniffing; examples of possible causes and consequences included unencrypted communication and data theft. As detection conditions, we included the presence of cleartext traffic, or anomalous network behaviour. As common patterns we highlighted automation rules that transmit sensitive data without encryption or react to unauthenticated inputs. Severity was classified as high.

These categories and the related examples were derived from reviewing the state of the art, in the area of IoT, End-User Development (EUD), and associated security and privacy concerns, such as [5],[6],[8],[14], and [16].

By explicitly focusing on these dimensions, our prompt aligns with current notions of responsible prompting [21], as it aims to proactively detect and mitigate potential harms to users’ security and privacy.

The prompt generates concise, non-technical warnings when a security or privacy concern is detected, offering explanations of the risk and possible mitigation strategies, while also suggesting safer alternative configurations. It ensures that warnings are shown only when relevant, prioritising high-severity issues and combining multiple concerns into a single, coherent message when appropriate. If the rule does not pose any significant risk, no warning is displayed. The completed prompt is publicly available².

The first implementation phase involved evaluating different prompts using the web version of ChatGPT. Multiple iterations of prompt engineering techniques were conducted to determine the most effective approach for guiding the model in automation creation and validation. This process was inspired by the guidelines

and prompting techniques outlined in a study [22], which provides a comprehensive overview of strategies for optimising prompts and improving language model responses.

Following the principles of prompt engineering, we started with a basic prompt, which was then refined through an iterative approach. As suggested in the literature, we introduced a quality control mechanism by asking metacognitive questions before the model carried out its task: (1) Do you need any additional information? (2) Are there any ambiguities in the prompt? (3) Do you need to ask me further questions to perform the task? (4) Would more examples help you better understand the problem? Based on the responses, we repeatedly adjusted and restructured the prompt. Specifically, we clarified how to handle cases involving multiple security issues - prioritising the one with the highest severity - and specified that the assistant should not only provide a warning but also a short explanation of the risk along with a mitigation suggestion.

The following step was testing different formats, varying the order of examples, and experimenting with different approaches to phrasing the instructions. Additionally, in line with the benchmarking and validation practices recommended [22], we conducted further testing by incorporating a smart home configuration (see Figure 1) and newly defined automations directly into the prompt.

To ensure the robustness of the generated responses, we then initiated a second ChatGPT session, asking the model to critically analyse its own outputs, assessing their correctness and coherence concerning security and privacy objectives.

Finally, the refined prompt is now able to evaluate whether an automation could lead to unintended security vulnerabilities (e.g., unauthorised access to home devices), identifies privacy concerns, and suggests mitigation strategies to users, helping them refine their automation rules before implementation.

3.2 A Typical Conversation

To illustrate the assistant’s functioning, consider the following example of a typical conversation with the chatbot. The *System* messages are not visible to the user, but in the following example are used to show the internal process of the system.

SecureTAP: *Hi! I’m your smart home assistant. I can help you create automations and make sure they’re safe and private. How can I help you?*

User: *I’d like to turn off the alarm when my phone connects to the home Wi-Fi.*

SecureTAP: *Got it. Let me break that down. Event: Your phone connects to the home Wi-Fi. Condition: (none specified - would you like to add one?). Action: Turn off the home alarm.*

Would you like to proceed with this rule as it is, or would you like to add any conditions (e.g., time of day, presence)?

User: *No conditions.*

System: The system generates the automation in JSON format. The generated automation is then passed to GPT-4o with the prompt presented in Section 3.1 to identify issues related to the automation and possible improvements.

SecureTAP: *Turning off the alarm automatically when your phone connects to Wi-Fi may pose a security risk. If your phone is lost or stolen, someone could gain unauthorized access to your home simply by being near your Wi-Fi network. Suggested improvement: add a*

²At the link: giove.isti.cnr.it/demo/shared/securetap_prompt.pdf

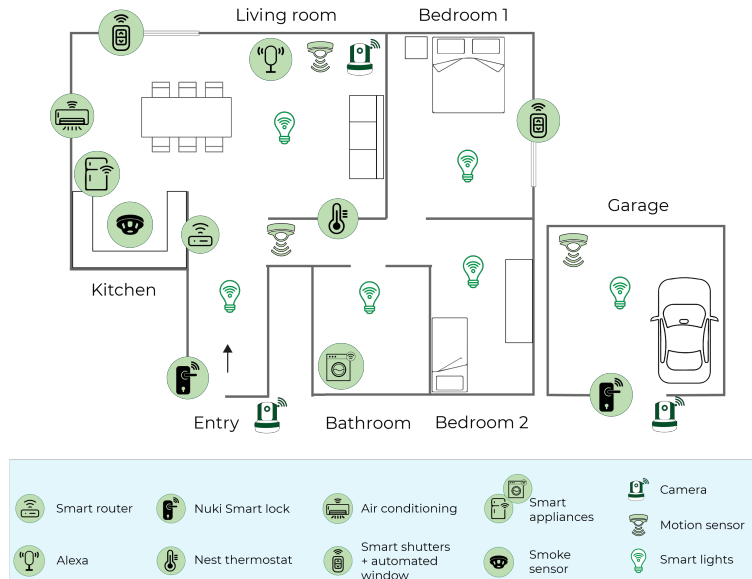


Figure 1: Planimetry of the configured Smart Home for the user test

second verification step, for example use biometric authentication, voice recognition, or allow the rule only during certain hours. Would you like to apply one of these improvements?

User: Yes, let's add biometric authentication for confirmation.

System: generates the modified automation considering the new user instruction. The modified automation is checked again to identify any new issues.

SecureTAP: Great! I've updated the rule to include biometric authentication before deactivating the alarm. Your automation is now safer and ready to use. Can I save the automation?

User: Yes.

System: Store the generated automation in the database.

4 Evaluation

4.1 Study Design

To evaluate SecureTAP, we carried out a user study involving fifteen participants. Each session lasted approximately 45 minutes and followed a structured process. Participants first received a brief introduction to IoT concepts and trigger-action programming, followed by a walkthrough of the smart home setup used in the study. This setup included a visual floor plan of the house with annotated sensors and smart devices (see Figure 1).

After this introduction, participants completed a demographic questionnaire, and a pre-test questionnaire aimed at assessing their initial understanding of security and privacy in smart home automations. They then created a SecureTAP account and were asked to create five automation tasks, starting from a scenario description, by interacting with the conversational assistant. Task instructions were presented one at a time, and the time taken to complete each task was recorded. These tasks were framed within realistic scenarios, such as automating the deactivation of an alarm when a known device connects to the home Wi-Fi network. Each of the

five scenarios was presented only once, to avoid repetition effects or memorisation. The full set of the scenarios used is the following:

SCENARIO 1: It's a hot summer day and the temperature in your home is starting to rise. To keep the environment cool without using the air conditioning, you decide to create an automation that automatically opens the living room windows when the temperature reaches 26°C, between 12:00 and 19:00. This way, the air circulates, and the house remains more comfortable.

SCENARIO 2: You want to make it easier for family members to access your home without having to use keys. You decide to create an automation: when a family member says "Unlock the door" to the voice assistant, the lock on the front door opens automatically.

SCENARIO 3: After a long day at work, you come home with your hands full of shopping bags. You decide to simplify the use of the home alarm system, avoiding having to deactivate it manually every time you return home. You then decide to create an automation so that, when a device registered on the network connects to the home Wi-Fi, the alarm is automatically deactivated.

SCENARIO 4: It's evening and you're away from home. You want to make sure that your home is always under control, even when you're not there. You then decide to create a new automation: if the security cameras detect movement outside when no one is home, the camera at the entrance takes a photo and sends it to all family members, via a notification on Telegram. This way, everyone can be informed in real time about any suspicious activity around the house.

SCENARIO 5: You want to improve safety and comfort in your kitchen and home. First, to quickly disperse smoke in case something goes wrong while cooking, you decide to create an automation: if smoke is detected near the oven for more than 30 seconds, the kitchen window opens automatically for 20 minutes, and a notification is sent via Telegram.

Additionally, to ensure windows are not left open unnecessarily at night, you create a second automation: every day at 20:00, all

windows in the house are automatically closed, regardless of the indoor temperature. If a window is reopened afterwards, it will be closed again automatically after 10 minutes.

Once all tasks were completed, participants filled in a post-test questionnaire, the Chatbot Usability Scale (BUS) questionnaire [2], and answered a set of additional questions regarding their experience with the system.

4.2 Participants and demographics

The study involved a balanced group of volunteers, comprising seven men and eight women, aged between 20 and 57 years (mean age 29.6, SD = 11.79). Participants were recruited informally through a call for volunteers and received no financial compensation. In terms of educational background, most participants (60%) held a bachelor’s degree, while 20% had completed a doctoral degree and the remaining 20% had finished high school. Programming experience varied: one participant reported having no experience, three described themselves as having low experience, six rated their skills as average, three as good, and two as very good. When asked about their familiarity with TAP, the majority (eight participants) indicated low experience, five reported a decent level of familiarity, and two described their experience as good.

4.3 Pre- and post-test questionnaire

The evaluation included a pre-test and post-test component to assess whether SecureTAP helped improve participants’ awareness of security and privacy risks in smart home environments. The pre-test focused on participants’ self-perceived ability to recognise and mitigate such issues, using a scale ranging from “not competent at all” to “very competent.” It also included two multiple-choice questions where participants were asked to identify potential problems—such as privacy violations, network vulnerabilities, or authentication errors—in given automation examples. In addition, participants were asked to write two automation rules: one to manage air circulation when temperatures became too high, and another to turn off all lights and lower the shutters when no one was home. These tasks were designed to explore participants’ practical understanding of TAP before using the assistant.

The post-test mirrored the structure of the pre-test, enabling a direct comparison of participants’ responses before and after the interactive session. The intentional repetition of questions and tasks was necessary to measure changes in participants’ understanding and answers over time. Indeed, this comparison was aimed at identifying any learning effect facilitated by SecureTAP.

4.4 Usability and User Experience

To assess the system’s usability, participants completed the BUS questionnaire [2], a standardised instrument for evaluating user satisfaction, perceived effectiveness, and ease of use with such technology. This questionnaire includes items related to how intuitive and efficient the system feels, as well as the overall quality of the user experience. Higher BUS scores reflect greater usability and user satisfaction. Following the BUS, participants answered several additional questions designed to gather qualitative insights. These included ratings of the usefulness of the assistant’s security suggestions and the extent to which their understanding of smart home

security risks had improved, both on a scale from 1 to 5. They were also invited to describe what they had learned from the experience, suggest improvements for the assistant, and indicate—on a scale from 1 to 10—how likely they would be to recommend SecureTAP to others as a tool for creating safer automations.

4.5 Results

4.5.1 Task completion time. For each user and for each task, we recorded the time taken for completion. The average time taken to complete the first task was 261 seconds (4.35 minutes); for the second task was 257 seconds (4.28 minutes); for the third task was 240 seconds (4 minutes); for the fourth task was 312 seconds (5.20 minutes); and for the fifth was 382 seconds (6.36 minutes).

The average completion time across all tasks for users who reported using ChatGPT multiple times a day was 5.53 minutes, whereas for those who indicated “rarely,” the average time was 7.35 minutes.

4.5.2 Comparison of pre-test and post-test responses. To evaluate the assistant’s impact on user awareness of security and privacy risks, we analysed responses from both the pre- and post-test questionnaires.

Self-perceived Competence in Identifying Security and Privacy Risks. In the pre-test, participants were asked to rate their own ability to identify and prevent security and privacy issues in smart home automations. Eight participants considered themselves “fairly competent”, five reported being “slightly competent”, and two “not competent at all”. After using the assistant, five participants reported that their competence had “greatly improved”, eight stated it had “moderately improved”, one reported a “slight improvement”, and only one participant reported “no improvement” (despite previously rating themselves as “fairly competent”).

Recognition of Privacy and Security Issues in Automations. Participants were shown the following automation: “WHEN motion is detected outside your home, THEN the security camera records and sends video to a cloud server” and asked to identify any potential issues. The correct answers were “privacy violation” and “network threats”. In the pre-test, 10 participants selected “privacy violation”, 4 selected “network threats”, 5 chose “authentication issues”, and 2 saw “no problem”. In the post-test, all participants correctly identified “privacy violation” as a concern; additionally, 3 selected “network threats” and 5 still indicated “authentication issues”. No participants selected “no problems”. These results indicate an increased awareness of privacy-related risks.

In a second example, participants were presented with: “WHEN my phone connects to my home Wi-Fi THEN it deactivates the alarm”. In the pre-test, 6 participants selected “I think there is a problem, but I can’t identify it”, one chose “no problem at all”, and 8 were able to articulate specific concerns (e.g., “my phone could be stolen”). After using the assistant, only 3 participants remained unsure, while 12 clearly identified relevant security or privacy issues, demonstrating improved analytical ability.

Ability to Write Secure Automations. Participants were asked to write an automation to ventilate the house when temperatures are too high. In the pre-test, all participants wrote syntactically correct automations, but only 1 included any form of safety condition (ensuring it’s not raining outside, and someone is

Table 1: BUS questionnaire results

ITEMS	AVERAGE VALUE
The chatbot functions are easily identifiable	4.27
The communication with the chatbot was clear	4.47
The chatbot was able to take into account what we said in the context of the conversation	4.4
The chatbot’s responses were easy to understand	4.6
I found that the chatbot understood what I wanted and helped me achieve my goals	4.27
The chatbot gave me adequate amount of information	4.4
The chatbot gave me only the information I needed	3.93
I think the chatbot’s responses were accurate	4.33
I believe the chatbot informed me of all the possible privacy issues.	4.43
The time I had to wait for the chatbot’s responses was short.	4.6

at home). In the post-test, 10 participants included some form of safety condition (such as ensuring someone is at home or adding a notification of the action), although not all were complete or fully accurate.

Similarly, another question required participants to write an automation that turns off all lights and lowers the blinds when nobody is at home. All users produced well-structured automations in the pre-test, but only one participant included security-related conditions (ensuring there are no movements for 5 minutes after the front door is closed). After using SecureTAP, three users explicitly included safety or security conditions (such as a telegram notification), and the overall quality of the automations improved: most participants rewrote their automations in a more complete and specific manner, demonstrating greater awareness of context and consequences.

4.5.3 User feedback and insights from testing. Participants were asked to reflect on their experience with the assistant and to evaluate its usefulness in promoting awareness of security and privacy risks in home automation. When asked how useful they found the security suggestions (on a 5-point Likert scale), all participants responded positively: nine selected “4”, five selected “5”, and one selected “3”. Regarding the question of whether their understanding of security risks improved after using the assistant, seven participants selected “4” and seven selected “5”, with only one participant selecting “1”. These results indicate that the assistant played a significant role in enhancing users’ awareness of potential vulnerabilities in smart home automations.

Participants were also invited to comment on what they had learned. Responses highlighted a range of insights, including: a heightened awareness of the importance of writing clear and precise automation rules; a better understanding of subtle or non-obvious risks; an appreciation for having a system that proactively suggests mitigation strategies rather than relying solely on user initiative; and a more comprehensive view of the issues that can emerge from home automation and IoT ecosystems.

To gauge user satisfaction, we collected Net Promoter Score (NPS) [24] ratings regarding how likely participants would be to recommend the system to others for improving automation security (on a scale from 1 to 10). The average score was 8.5. Specifically, five participants selected “8”, four selected “10”, four selected “9”,

one selected “7”, and one selected “4”. This distribution suggests a high level of satisfaction and perceived value.

Finally, participants provided suggestions for improvement. Several praised the assistant’s simplicity and intuitiveness, while others proposed enhancements such as the inclusion of visual elements (e.g., icons, images, emojis) to improve interpretability of suggestions; support for multimodal interaction (e.g., voice input/output); broader coverage of possible security risks; and more effective handling of rule conflicts.

4.5.4 BUS questionnaire results. The BUS questionnaire allowed us to determine the system’s usability and user satisfaction. In the table below, we summarise the user responses to each question, with the average value (on a scale from 1 to 5).

5 Discussion

User testing provided an important validation of the prototype’s usability and learning potential. The results of the BUS questionnaire were positive, indicating that users perceived the system as both usable and satisfying. This feedback supports the effectiveness of our design in delivering a clear and engaging user experience.

The comparison between pre-test and post-test answers suggests a fair learning effect. After using SecureTAP, several users showed improved understanding of trigger-action programming, particularly in recognising the potential consequences of automation in terms of security and privacy. This learning effect was also evident in users’ behaviour: starting from the second task, many participants spontaneously introduced safety conditions, trying to create a more complete and secure automation, reflecting an increased awareness of risks and a proactive attitude toward secure rules. To assess whether these automations became objectively more secure, we focused on indicators of risk mitigation, such as the inclusion of safety conditions (e.g., presence detection, notifications and confirmation steps, etc.). The increase in such conditions—from 1 to 10 participants in one task and from 1 to 3 in another—was interpreted as a proxy for improved security awareness and practice.

Participants’ open-ended responses to the question “What have you learned?” reinforced these findings: many participants emphasised the importance of being clearer and more precise when creating automations and how this could help in preventing risks. Several users noted that they had learned to consider potential

risks, such as privacy violations or security vulnerabilities, before creating an automation. Some participants highlighted that they had gained new insights into the importance of writing complete and thoughtful rules, considering all possible scenarios, including edge cases that may not be immediately obvious. Overall, responses showed an increased understanding of privacy and security principles, and also a shift in users' mindset toward a more preventive and structured approach to automation design.

These findings also help rule out that the positive results (from the BUS, NPS, and open-ended comments) were driven by participants' desire to please the experimenters, as their behaviour (such as spontaneously introducing safety conditions without prompting) and their open-ended responses demonstrated genuine learning about security and privacy risks.

While a large portion of participants (8) reported little to no prior exposure to TAP - responding into the questionnaire with "no experience," "never programmed," or "only recalling mentions of it in university settings" - the remaining 7 users mentioned experience in university projects, or commercial tools like Alexa. This variety in background offers useful insight: although more experienced users appeared to complete tasks more quickly, even those with no prior experience were able to understand the system and learn from it, confirming the accessibility and educational potential of the prototype.

The Net Promoter Score (NPS) result further supports the positive reception of the prototype, with a mean score of 8.5. This suggests that most of the users found the system not only usable but also worthy of recommending to others.

Areas for improvement. Participants also pointed out areas for improvement. When asked what they would change or improve in SecureTAP, some answered "nothing" or explicitly stated that it was already intuitive and easy to use. However, other responses suggested potential enhancements, such as supporting voice input/output for faster interaction, simplifying explanations, and including more visual elements like images or emojis to aid understanding. Some participants also recommended alternative interaction methods, such as using buttons to compose automation rules.

An issue that we noted was that SecureTAP sometimes reported an excessive number of potential security problems, even when users had already included security conditions in their automation rules, resulting too verbose. This redundancy, although intended to offer additional safeguards, could become frustrating for users who felt they had already addressed potential risks. This extra feedback might seem excessive, detracting from the overall user experience and making the time to create and save an automation too long. This aspect is also reflected in the BUS questionnaire: question number 7 ("the chatbot gave me only the information I needed") obtained the lowest value of 3.9.

Moreover, in some cases, users did not follow SecureTAP's suggestions, preferring to trust their own reasoning or found the assistant's feedback not entirely helpful. Probably because, in these cases, the suggestion often involved notifications or multiple additional actions which, although conceptually valid, were perceived as excessive or intrusive.

6 Conclusion and Future Work

This study introduced and evaluated a conversational agent designed to support the creation of smart home automations with an explicit focus on security awareness. User testing and feedback confirmed that the prototype substantially meets its core objectives of promoting usability and fostering attention to security-related concerns. The system was perceived as engaging, intuitive, and easy to use, as reflected by the positive BUS questionnaire results and a high Net Promoter Score. Additionally, the observed learning effect suggests that the system can enhance users' understanding of Trigger-Action Programming, especially in relation to security and privacy issues. Notably, even participants without prior experience in TAP were able to grasp the concepts and appreciate the importance of anticipating potential risks when creating automations.

Future work will aim to refine the assistant based on these insights to provide a more accurate and concise chatbot feedback. Moreover, we plan to apply the chatbot in real smart home settings for longer-term validation, to further assess and improve its effectiveness.

Acknowledgments

This work was partially supported by the project SERICS (PE0000014) under the MUR National Recovery and Resilience Plan funded by the EU NextGeneration EU projects. The support is gratefully acknowledged.

References

- [1] J. Abbott, J. Dev, D. Kim, S. Gopavaram, M. Iyer, S. Sadam, S. Mare, T. Ringenberg, V. Andalibi, and L. Jean Camp. 2022. Privacy Lessons Learnt from Deploying an IoT Ecosystem in the Home. 2022. 98–110. <https://doi.org/10.1145/3549015.3554205>
- [2] Simone Borsci, Elisa Prati, Alessio Malizia, Martin Schmettow, Alan Chamberlain, and Stefano Federici. 2023. Ciao AI: the Italian adaptation and validation of the Chatbot Usability Scale. *Pers Ubiquit Comput* 27, 6 (December 2023), 2161–2170. <https://doi.org/10.1007/s00779-023-01731-2>
- [3] Bernardo Breve, Gaetano Cimino, and Vincenzo Deufemia. 2024. Hybrid Prompt Learning for Generating Justifications of Security Risks in Automation Rules. *ACM Trans. Intell. Syst. Technol.* 15, 5 (November 2024), 103:1–103:26. <https://doi.org/10.1145/3675401>
- [4] A.J. Bernheim Brush, Bongshin Lee, Ratul Mahajan, Sharad Agarwal, Stefan Saroiu, and Colin Dixon. 2011. Home automation in the wild: challenges and opportunities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*, May 07, 2011. Association for Computing Machinery, New York, NY, USA, 2115–2124. <https://doi.org/10.1145/1978942.1979249>
- [5] Lei Bu, Qiuping Zhang, Suwan Li, Jinglin Dai, Guangdong Bai, Kai Chen, and Xuandong Li. 2023. Security Checking of Trigger-Action-Programming Smart Home Integrations. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2023)*, July 13, 2023. Association for Computing Machinery, New York, NY, USA, 639–651. <https://doi.org/10.1145/3597926.3598084>
- [6] Xuyang Chen, Xiaolu Zhang, Michael Elliot, Xiaoyin Wang, and Feng Wang. 2022. Fix the leaking tap: A survey of Trigger-Action Programming (TAP) security issues, detection techniques and solutions. *Computers & Security* 120, (September 2022), 102812. <https://doi.org/10.1016/j.cose.2022.102812>
- [7] Fulvio Corno, Luigi De Russis, and Alberto Monge Roffarello. 2021. From Users' Intentions to IF-THEN Rules in the Internet of Things. *ACM Trans. Inf. Syst.* 39, 4 (August 2021), 53:1–53:33. <https://doi.org/10.1145/3447264>
- [8] Earlece Fernandes, Jaeyeon Jung, and Atul Prakash. 2016. Security Analysis of Emerging Smart Home Applications. In 2016 IEEE Symposium on Security and Privacy (SP), May 2016. 636–654. <https://doi.org/10.1109/SP.2016.44>
- [9] Simone Gallo, Fabio Paterno, and Alessio Malizia. 2023. Conversational Interfaces in IoT Ecosystems: Where We Are, What Is Still Missing. In *Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia (MUM '23)*, December 03, 2023. Association for Computing Machinery, New York, NY, USA, 279–293. <https://doi.org/10.1145/3626705.3627775>

- [10] Simone Gallo, Fabio Paternò, and Alessio Malizia. 2024. A conversational agent for creating automations exploiting large language models. *Pers Ubiquit Comput* 28, 6 (December 2024), 931–946. <https://doi.org/10.1007/s00779-024-01825-5>
- [11] Weijia He, Jesse Martinez, Roshni Padhi, Lefan Zhang, and Blase Ur. 2019. When Smart Devices Are Stupid: Negative Experiences Using Home Smart Devices. In *2019 IEEE Security and Privacy Workshops (SPW)*, May 2019. 150–155. <https://doi.org/10.1109/SPW.2019.00036>
- [12] Victor Jüttner, Arthur Fleig, and Erik Buchmann. 2024. ChatAnalysis: Can GPT-4 undermine Privacy in Smart Homes with Data Analysis? 2024. Gesellschaft für Informatik e.V., 10.18420/muc2024. Retrieved March 31, 2025 from <https://dl.gi.de/handle/20.500.12116/44282>
- [13] Bhavyasri Kadali, Neha Prasad, Pranaya Kudav, and Manoj Deshpande. 2020. Home Automation Using Chatbot and Voice Assistant. *ITM Web Conf* 32, (2020), 01002. <https://doi.org/10.1051/itmconf/20203201002>
- [14] Rohan Kar and Rishin Haldar. Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements. ResearchGate. Retrieved March 11, 2025 from https://www.researchgate.net/publication/310122692_Applying_Chatbots_to_the_Internet_of_Things_Opportunities_and_Architectural_Elements
- [15] Fu Li, Jiaming Huang, Yi Gao, and Wei Dong. ChatIoT: Zero-code Generation of Trigger-action Based IoT Programs with ChatGPT | Proceedings of the 7th Asia-Pacific Workshop on Networking. Retrieved March 24, 2025 from <https://dl.acm.org/doi/abs/10.1145/3600061.3603141>
- [16] Phillip L. Morgan, Emily I. M. Collins, Tasos Spiliotopoulos, David J. Greeno, and Dylan M. Jones. 2022. Reducing risk to security and privacy in the selection of trigger-action rules: Implicit vs. explicit priming for domestic smart devices. *International Journal of Human-Computer Studies* 168, (December 2022), 102902. <https://doi.org/10.1016/j.ijhcs.2022.102902>
- [17] Muhamad Muslih, Somantri, Dedi Supardi, Elpid Multipl, Yusup Maulana Nyaman, Aditya Rismawan, and Gunawansyah. 2018. Developing Smart Workspace Based IOT with Artificial Intelligence Using Telegram Chatbot. In *2018 International Conference on Computing, Engineering, and Design (ICCED)*, September 2018. 230–234. <https://doi.org/10.1109/ICCED.2018.00052>
- [18] C. Norval and J. Singh. 2024. A Room with an Overview: Toward Meaningful Transparency for the Consumer Internet of Things. *IEEE Internet of Things Journal* 11, 5 (2024), 7583–7603. <https://doi.org/10.1109/JIOT.2023.3318369>
- [19] Bauyrzhan Ospan, Nawaz Khan, Juan Augusto, Mario Quinde, and Kenzhegali Nurgaliyev. Context Aware Virtual Assistant with Case-Based Conflict Resolution in Multi-User Smart Home Environment. In *ResearchGate*, . <https://doi.org/10.1109/CoCoNet.2018.8476898>
- [20] Ahmed Sakib, Paul Debashish, Masnun Rubaiya, Ahmed Shanto Minhaz Uddin, and Farah Tanjila. Smart Home Shield & Automation System Using Facebook Messenger Chatbot. In *ResearchGate*, . <https://doi.org/10.1109/TENSYMP50017.2020.9230716>
- [21] Vagner Figueredo De Santana, Sara Berger, Tiago Machado, Maysa Malfiza Garcia de Macedo, Cássia Sanctos, Lemara Williams, and Zhaoqing Wu. 2025. Can LLMs Recommend More Responsible Prompts? March 24, 2025. . Retrieved May 15, 2025 from <https://research.ibm.com/publications/can-llms-recommend-more-responsible-prompts>
- [22] Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncareanu, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Hoyle, and Philip Resnik. 2025. The Prompt Report: A Systematic Survey of Prompt Engineering Techniques. <https://doi.org/10.48550/arXiv.2406.06608>
- [23] V. Zimmermann, P. Gerber, K. Marky, L. Böck, and F. Kirchbuchner. 2020. Assessing users' privacy and security concerns of smart home technologies. *i-com* 18, 3 (2020), 197–216. <https://doi.org/10.1515/icom-2019-0015>
- [24] Net Promoter System®. *Bain*. Retrieved May 14, 2025 from <https://www.netpromotersystem.com/>