# An Empirical Study on the Effectiveness of Privacy Indicators

Michele Guerra, *Student Member, IEEE,* Simone Scalabrino, Fausto Fasano, and Rocco Oliveto, *Member, IEEE*

**Abstract**—The increasing diffusion of mobile devices and their integration with sophisticated hardware and software components has promoted the development of numerous applications in which developers find new ingenious ways to exploit the possibilities offered by the access to resources such as cameras, biometric sensors, and GPS receivers. As a result, we are increasingly used to seeing applications that make extensive use of sensitive resources, potentially dangerous for our privacy. To address this problem, the latest approach to support user awareness in terms of privacy is represented by the Privacy Indicators (PI), a software solution implemented by the operating system to provide a visual stimulus to inform users whenever a dangerous resource is exploited by the app. However, the effectiveness of this approach has not been assessed yet.

In this article, we present the result of a study on the effectiveness of using the PI to inform the user every time an app accesses the mobile device camera or microphone. We have chosen these two resources as the PI are currently implemented only for a very limited number of permissions. The controlled experiment involved 122 Android users who were asked to complete a series of tasks on their smartphone through prototypes using the involved resources in an explicit and latent way. Although the PI mechanism is very similar between Android and iOS, we have decided to focus on the former due to its greater diffusion. The results show no significant correlation between the use of PI and the detection of the resource being used by the app, suggesting that the effectiveness of PI in improving sensitive-related resources usage awareness, as currently implemented, is still unsatisfactory. In order to understand if the problem was due to the specific implementation of the PI, we implemented an enhanced version and compared it with the standard one. The results confirmed that an implementation that makes the indicators more visible and that is clearer in highlighting the fact that the app is accessing a resource improves the resources usage awareness.

**Index Terms**—Android Permissions, Privacy, Empirical evaluation.

## 1 Introduction

Modern mobile devices are shipped with increasingly advanced hardware and software components capable of integrating and extending the possibilities of the device. Managing the access to these resources represents a technological challenge for OS vendors. In fact, many of these components produce or manage sensitive information, which could be misused to profile users or even spying on them.

Examples of such privacy-sensitive hardware components are the camera, the gyroscope, the biometric sensors, the microphone, the GPS receiver, and the external storage, while software components that can represent a threat to privacy include the media library, the address book, the phone call, and SMS manager.

Many applications need to legitimately access one or more of these resources in order to provide advanced functionalities or to ensure a better user experience. Therefore, a mechanism to control the access to the available resources must be provided.

The management of access permissions to sensitive resources of modern mobile devices is entrusted to a particular component of the operating system whose task is to mediate the requests of each App according to the preferences specified by the user. For example, in Android to read the SMS within the cellphone SMS list, the App must obtain the READ_SMS permission.

Unfortunately, the decision to allow or deny access to a system resource cannot be entirely delegated to the operating system. Indeed, even if a generic clue can be found in the app category – *e.g.,* we expect that a voice recording app must be allowed to access the microphone and that in order to scan a QR code an app should be granted access to the camera – it is not possible in advance to establish if and when a specific app should be allowed to access a resource. Furthermore, it has been highlighted that even popular apps often use the same resource within different usage contexts [1]. For this reason, the choice is generally left to the end user.

Over the years, there have been many improvements to the authorization system, which has increased the perceived level of security, especially for what concerns the user privacy protection. The reasons why we still cannot rely on an effective mechanism to protect users from possible violations of their privacy are various, including: the need to offer a simple and effective solution, where the user is not continuously interrupted by the authorization system – often without even knowing the effects of one choice over another – which increases user fatigue; and the timeliness of controls, *i.e.,* user do not want to interrupt the regular use of the App, while maintaining temporal proximity between the access request and the actual use of a resource [2].

Starting from Android 12, new improvements have been

• *M. Guerra, S. Scalabrino, F. Fasano and R. Oliveto are with the University of Molise, Italy*
*E-mail: [michele.guerra, simone.scalabrino, fausto.fasano, rocco.oliveto] @unimol.it*

*Manuscript received April 19, 2005; revised August 26, 2015.*

introduced to the permission model regarding the use of resources and the user awareness. In particular, Privacy Indicators (PI) have been introduced, which are notification indicators aimed to inform the user, through a colored dot on top of the screen, about the use of a device resource, such as the camera or the microphone (Figure 1).

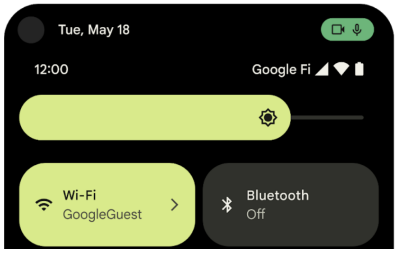



Fig. 1. An example of Privacy Indicator in action on Android 12.

This mechanism could be a valid compromise between privacy awareness and non-intrusiveness, since the user is not continually interrupted by the system, but is nevertheless warned about the resource usage promptly. However, while ensuring an adequate user experience, there is a risk that users do not identify and recognize such warnings effectively.

Although there is evidence that attention can operate on or be drawn to unconscious stimuli, various studies question on awareness without attention in case of multiple visual stimuli [3]. In the majority of cases, objects or data relevant to the task can be identified in advance. During the actual visualization, the viewer's visual system must focus its attention on these objects and data in order to complete the task [4]. Thus, when participants are focused on the primary task, a new and unexpected stimulus could remain unnoticed. This is known as inattentional blindness [5].

In this paper, we investigate the effectiveness of PI on users' perception of resource utilization. Our study aimed at answering the following Research Question:

**RQ1** *How effective are Privacy Indicators to help users identifying when a resource is used by a mobile application?*

To this aim, we planned and conducted a controlled experiment with 122 participants to explore the effectiveness of PI in making users aware of resource usage in two contexts: *latent* and *explicit*. In *latent* contexts, resources – camera and microphone, in our study – are accessed even if they are not necessary for the task and the apps do not present any preview or animation on the device display while using a resource, except for the PI. In *explicit* contexts, apps actually need to access the resource to complete the task. Moreover, the expected behaviour is implemented when using a resource, such as camera preview or microphone icon animation. We designed four tasks in which developers were asked to perform actions on four popular Android apps, *i.e.*, WhatsApp, Facebook Messenger, Spotify, and Twitter. Each task was administered with two treatments, *i.e.*, visible or non-visible PI, leading to eight possible combinations. To run the experiment, we implemented eight interactive prototypes: Such prototypes aim at simulating the Android OS and guiding the participants through the execution of the tasks while allowing us to fully control the PI visibility and other possible confounding variables (*e.g.*, notifications).

We performed statistical analysis to answer RQ1, and our results show that the association between the presence of PI and the identification of resource usage is not statistically significant. Besides, the effect size is very low. This demonstrates that, while a slight increase in identification can be observed in some cases when PI are shown to the user, such a feature is still not entirely adequate to make users aware of camera and microphone usage. Motivated by these findings, we introduced an enhanced implementation of privacy indicators aiming to overcome the limitations identified with the standard PI. Thus, we extended our study to investigate if users' awareness of app resource usage can be increased, leading to a new research question:

**RQ2** *Do enhanced indicators improve user ability to identifying when a resource is used by a mobile application compared to standard PI?*

We used the same design we previously adopted to answer RQ1 while changing the treatments, which this time included (i) Android PI and (ii) our enhanced PI. This time, we involved 38 participants. Our results clearly show that users are significantly more aware of resource usage when using our enhanced PI. This shows promise for alternative strategies that could further enhance such a privacy features introduced in Android 12. To summarize, the main contributions of this paper are:

- We investigate how effective are Android privacy indicators to make users aware of resource usage. Our results show that the currently used version of PI does not significantly impact the user's awareness;
- We assess the effectiveness of a more prominent and informative privacy indicator aimed at increasing the user's awareness of resource usage. Our results show that, compared to Android PI, such an alternative visualization significantly improves the user's awareness.

Android maintainers can benefit from our work because they will obtain evidence about users' consciousness of privacy indicators and how to possibly improve such an important feature.

## 2 Android's Permission Model Evolution

Initially, the Android's permission model was too simple to ensure user privacy properly. When Android 1.0 was released, mobile apps were allowed to access any resource without the need to declare it. The permission model in Android 3.0 was enhanced to prevent apps from accessing external storage. In Android 4.4, apps started to declare permission during the installation phase, and version 5.0, further improved the permission model adding new permissions to the list, but still at install-time.

These first versions of Android did not actually implement an access control management. Users could either grant all of the requested permission or reject the request

and quit the install procedure. In case the user had agreed to proceed with the installation, the app could have used these resources at any time.

This clearly posed a problem because there was no way of knowing if and when the application was accessing a sensitive resource without the user's awareness.

Even if this first approach had undoubted advantages related to the natural selection of applications that were not hungry for permissions and the non-intrusiveness of the privacy management system during regular app use, all these approaches proved to be too rigid and ineffective in addressing the problem [6]. It has been observed that inadequately contextualized requests (such as those made at install-time) do not allow the user to understand the actual risk involved in granting or denying permission [2]. For instance, Felt et al. [6] examined whether the Android permission system is effective at warning users. They evaluated whether Android users pay attention to, understand, and act on permission information during installation performing an internet survey and a laboratory study. They found that current Android permission warnings do not help most users make correct security decisions.

In order to mitigate unintended malicious behavior, Android's permission mechanism has moved from install-time to run-time since version 6.0, named Marshmallow, launched in 2015. In the run-time permission model, a request dialog is shown as soon as the app requests the access to a specific permission related resource, allowing the user to decide whether granting or denying that permission. The user decision can be stored by the operating system to avoid further requests for the same permission in subsequent usages. In case the app needs to use a different resource, a new permission request will be presented to the user. Even with the new access control mechanism, the operating system's ability to avoid undesired or ambiguous behaviors proved to be insufficient [7], especially due to the fact that, once a permission has been granted, the user looses any control over further accesses to the resource.

Successive systems have emphasized two features: contextualization and granularity of permissions. The former aims to enabling a more informed choice by delaying the decision until the application actually uses the permission or requests it. The latter allows the user to discern between an increasing number of permissions to better profile the user's viewpoint on privacy. Although both features are aimed at more detailed privacy management, there are still many weaknesses, and an effective solution is still not available.

Shen et al. [8] found out, through analysis of real users permission settings and through large-scale user studies, that users have several common misunderstandings about the use of specific permissions and that many Android users are unaware of changes in the permission model.

Finally, starting from Android API 31, named Android 12, released by Google in late 2021, new improvements have been introduced to the permission model with the introduction of Privacy Indicators and a dashboard aimed at enhancing the user awareness on privacy.

## 3 Empirical Study Design

The *goal* of our empirical study is to investigate whether PI can properly notify Android users about the use of a resource such as a camera or a microphone. Moreover, we aim to detect differences in identifying the use of the two resources between different contexts and whether one resource is more prominent than the other. The following research question guided our study:

**RQ1** *How effective are Privacy Indicators to help users identifying when a resource is used by a mobile application?*

We conducted a controlled experiment with human participants to answer this RQ1.

### 3.1 Context Selection

The context of our study is composed of *objects*, *i.e.,* apps to use in a simulated Android environment, and *subjects*, *i.e.,* human participants representing common Android users.

As for the objects, our objective was to select four popular Android apps based on which we could create tasks that participants could complete in a few minutes. We chose two messaging apps (WhatsApp and Facebook Messenger), a music app (Spotify), and a social network app (Twitter). We selected these apps because they are among the top 30 most popular free apps in the Google Play Store[1]. More specifically, WhatsApp and Messenger are ranked first and second in the *Communication* category[2], Spotify is the most downloaded in the *Music And Audio* category[3], and Twitter is ranked fifth in the *Social* category[4]. Our choice of popular and trusted apps for the experiment was based on the assumption that users are more inclined to grant full resource usage permissions to these apps. Typically, users might restrict camera or microphone permissions for less trusted apps to one-time use, whereas they might provide full access to apps they regularly use and trust, such as those employed in our study. We assume that the role of privacy indicators is particularly relevant in the context of these frequently-used and trusted apps.

We involved a total of 149 subjects (participants) in our study. A subset of them (27) were involved in a pilot study we run to test our experimental procedure (more on this later), while the remaining 122 were involved in our main experiment. Since the target population of our study is the whole population of Android users, we had no particular requirement in terms of skills or knowledge. We only made sure that all of them used an Android device as their primary phone. We made this choice because permission management among different OSs can be different and this could have been a bias for the experiment. The information obtained is recorded by the investigator in such a manner that the identity of the human subjects cannot readily be ascertained, directly or through identifiers linked to the subjects.

1. https://www.androidrank.org/android-most-popular-google-play-apps?price=free
2. https://www.androidrank.org/android-most-popular-google-play-apps?category=COMMUNICATION
3. https://www.androidrank.org/android-most-popular-google-play-apps?category=MUSIC_AND_AUDIO
4. https://www.androidrank.org/android-most-popular-google-play-apps?category=SOCIAL

### 3.2 Experimental Procedure

We considered three independent variables. The first one represents the support of Privacy Indicators at OS-level (PI): such a variable allows us to simulate both the Android versions lower than 12 (without PI) and greater than or equal to 12 (with PI). The second variable is the Context Behavior (CB), *i.e.,* the way in which the app uses the resources (camera and microphone): an app might use a resource either in a legitimate way (explicit) to provide a feature to the end user (*i.e.,* taking a picture or registering a voice message) or in a latent way (non-explicit) if the resource is not strictly required, but the app uses it anyway. The last variable is the resource used (R): as previously mentioned, the PI are shown when one of the two resources is used, *i.e.,* camera and microphone. In our experiment, PI is the factor, with two treatments (*i.e., visible* — pre-Android 12 — and *not visible* — post-Android 12). CB and R are co-factors we consider to instantiate the tasks. We take such co-factors into consideration to observe to what extent the treatments are effective in different contexts. The dependent variable is the correctness in the identification of cases in which an app uses a sensitive resource. Such a variable has two possible values: *true* (if the app uses a resource and the participant is aware of this) and *false* (otherwise). If the treatment in which PI are used works, we should observe a higher percentage of participants' awareness.

#### 3.2.1 Task Definition and Group Assignment

Given the four selected apps, we defined four tasks by combining the values of the two co-factors (CB and R). Specifically, the tasks are the following:

- $T_1$ [CB = Latent; R = Camera] The participant has to open the Twitter app and create a new tweet. A tweet composition window appears, with the message content already written, and the participant has to publish it by tapping the "Publish tweet" button. We simulate that the camera is used while the participant is composing the tweet.
- $T_2$ [CB = Explicit; R = Camera] The participant has to open the WhatsApp app and look for a chat with unread messages. Then, the participant has to take a picture and send it as an attachment.
- $T_3$ [CB = Explicit; R = Microphone] The participant has to open the Facebook Messenger app and search for a chat with a specific friend. The participant has to start a voice call with such a friend and close the call after a few seconds.
- $T_4$ [CB = Latent; R = Microphone] The participant has to open the Spotify app and search for a specific playlist from the list. A list of songs appears, and the participant is asked to play a specific song. We simulate that the microphone is used while the participant is looking for the song from the list.

A first requirement for our experiment was to make sure that it was feasible to execute it by many participants. Therefore, we wanted them to use their own smartphones, without asking to install apps or modified Android versions. Also, most importantly, we wanted to have the opportunity to manipulate the PI variable for all of them to administer the two treatments. To meet such requirements, we decided to create simulations of the Android OS that (i) guided the participants during the task execution, (ii) allowed us to explicitly show or hide PI, and (iii) could be accessed from any smartphone through the web browser, without specific requirements. We used Figma to create such simulations. Figma (similarly to others like Sketch and Adobe XD) is typically used to define high-fidelity prototypes of mobile apps, but we used it to prototype the entire Android OS (and the apps used for the respective tasks), by simulating it as closely as possible.


Fig. 2. An example of prototypes created using Figma.

Figure 2 shows two screenshots of prototypes created with Figma. It is worth noting that they accurately reproduce the actual apps user interface and behaviour, also simulating the interactivity of these apps. For the implementation of the prototypes, we assumed that the permissions for camera and microphone had been granted by the user in a previous use of the app. Such a choice was made to avoid that the system popup asking for permission granting would affect the participant answer about the usage of a resource, which is typical in the normal usage of an app. Also, we decided to limit the interactions that users can do on the prototypes; in this way, we reduced the likelihood that participants got stuck during the steps to be performed.

We defined two groups of participants. We asked each group to complete the same four tasks in the same order. The only difference was the PI value for each task: In the first group, T1 and T3 were assigned with the PI visible treatment, while T2 and T4 with the PI not visible treatment; in the second group, we assigned complementary treatments (PI visible for T2 and T4 and PI not visible for T1 and T3). Therefore, in total, we prepared eight prototypes. We summarize the groups and the values of the three independent variables we consider in Table 1. Each participant was randomly assigned to one of the two groups.

TABLE 1
Tasks and values assigned to the PI, CB, and R variables for each group.

| Group 1 | PI | CB | R |
|---|---|---|---|
| *Task 1* | visible | latent | camera |
| *Task 2* | not visible | explicit | camera |
| *Task 3* | visible | explicit | microphone |
| *Task 4* | not visible | latent | microphone |

| Group 2 | PI | CB | R |
|---|---|---|---|
| *Task 1* | not visible | latent | camera |
| *Task 2* | visible | explicit | camera |
| *Task 3* | not visible | explicit | microphone |
| *Task 4* | visible | latent | microphone |

#### 3.2.2 Experimental Protocol

The experiment consisted of 5 phases. In the *preliminary phase*, we acquired basic information about the participants; in the *learning phase*, we instructed them on the PI feature introduced in Android 12; in the *task execution phase*, the participants were asked to complete one of the tasks; finally, in the *response acquisition phase*, we collected information that allows us to compute the dependent variable, while in the *matching phase* we allowed participants to review their response based on static screenshots of the app. We provide an overview of the phases in Figure 3 and we report below the details for each of them.

The experiment was conducted using two devices. Participants were required to use a laptop/tablet and a smartphone concurrently. On the laptop/tablet, participants interacted with a web application that guided them through the entire experiment, presenting the instructions, tasks steps to follow, and questionnaires. The tasks, tied to smartphone use and PI, were performed on the participants' personal smartphones. During the experiment, two of the authors and two collaborators monitored the participants to ensure they completed the tasks independently and correctly.

**Preliminary phase.** In this phase, we administered a pre-questionnaire to the participants through the webapp. We aimed at acquiring both personal information (age, experience in terms of number of years they were using smartphones, instruction level, knowledge about smartphone operating systems, if they are specialized in computer engineering or computer science and their concern for privacy) and technical information (smartphone model, Android version installed).

**Learning/informative phase.** This phase aimed at making sure that all the participants were fully aware of the PI feature. We did not want to explicitly instruct the user about PI since there would have been a risk that they unnaturally focus on them during the tasks. Thus, we also instructed them about other Android 12 features (*e.g.,* privacy dashboard).

We first asked the participants to read from their laptop/tablet the Google official page that contains the features list with visual examples of them and asked to take time to navigate through the documentation. Then, we demonstrated such features through an interactive prototype (similar to the ones they would later use for the tasks), which was visualized directly on the users' smartphones, effectively simulating the behavior of Android 12. This provided the participants with a realistic experience of using the new features before proceeding with the tasks. At the end, we asked them if they fully understood those features, forcing them to start over in case of a negative answer.

After that, participants had to complete a guided task to get familiar with the prototypes with which they would later complete the tasks. We preliminarily informed the participants on how the prototypes work (*e.g.,* not all the OS features are available) and we explicitly told them they could assume that microphone and camera permissions in the simulator had been granted for any of the apps used during the tasks. Then, we asked them to use their smartphone for accessing a tutorial with a guided task similar to the actual ones (next phase). During this phase, participants were also presented with a mnemonic survey consisting of three questions related to the actions performed during the guided task and the elements present in the UI. This survey was introduced to further familiarize participants with the environment and the tasks they would later encounter. In this way, we tried to reduce the influence of learning effect. Indeed, in case we let the participants start completing the tasks, they would not have been fully aware of the process and, most of all, of the questions that would have been asked after the task. As a result, all the participants could perform less well on the first task.

**Task execution phase.** Participants were asked to complete the four tasks, with treatments assigned based on their groups. For each task, we showed on the webapp a QR code that linked to the prototype implementing the task with the given treatment. Then, the webapp (on the laptop/tablet) provided the instructions on the actions to perform and participants executed them on their smartphones. The prototypes provided a verification code to the participants at the end of each task that they had to insert in the supporting webapp on the laptop to proceed with the next phase. We did this to make sure that participants completed the tasks before answering the questions.

**Response acquisition phase.** The webapp showed a questionnaire at the end of every task, which included a control question and two questions about the resource usage. The first one was designed to ensure that participants were not carelessly completing the tasks and were, instead, maintaining focus. Note that such questions were not used in our design (*i.e.,* we do not use them for data analysis), but rather to reinforce participants' attention during the tasks. These questions were task-specific and regarded the actions performed (*e.g.,* what was the name of the person to whom you sent a message?). As a second question, we asked whether the app used any resources (*i.e.,* camera or microphone). If the answer was affirmative, we asked to indicate at which point the resources were used (*e.g.,* while writing the message). We later use the answer to the second and third questions to measure the dependent variable.

**Matching phase (only for tasks PI = visible).** In this last phase, we showed participants some screenshots of the tasks completed and we asked again the questions from the previous phase. One of the screenshots included the screen in which the PI was visible. This phase allows us to understand if there are differences between the answers provided only based on the information acquired while performing the tasks (*e.g.,* the participant did not pay enough attention to the PI because he/she was doing something else) and the ones provided while not performing a task. During this last phase, the user could confirm the answer given during the previous step by moving on, or change their answer by selecting from the proposed screenshots the one that is correct for them.

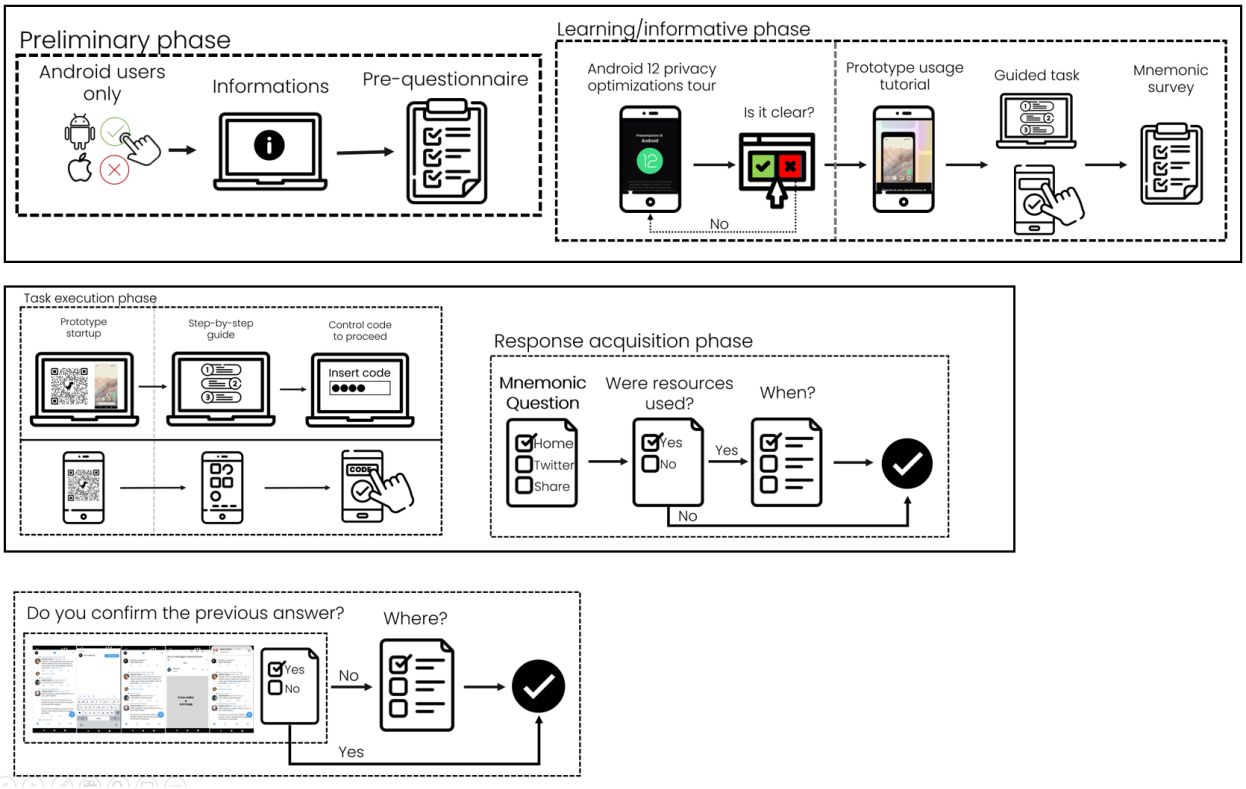



Fig. 3. Experimental protocol.

#### 3.2.3 Pilot Study

Before running our experiment, we conducted a pilot study with 27 participants in an offline setting. Such a study was promoted from December 29, 2021, to January 14, 2022. We tested the previously-described experimental protocol and we checked whether the tasks were feasible, whether the instructions were sufficient and explicit, whether the prototypes worked on different devices and their usability, and whether users experienced problems or had doubts during execution. None of the users reported problems in understanding and completing the tasks.

### 3.3 Data Analysis

The values of the independent variables we are interested in are determined based on the task and the group. We compute the value of the dependent variable based on the answers provided in the response acquisition phase. Specifically, if the participants answered "yes" to the second question (*i.e.,* they noticed that the app used a resource) and if they correctly identified the moment the resource was used (third question), the value of the dependent variable was *true*, otherwise it was *false*.

To answer our RQ1, *i.e.,* to determine whether there is a statistically significant association between the presence of PI and the identification of resource use, we use a statistical independence test. Such tests are used to determine if there is a significant relationship between two categorical variables. Specifically, we use Fisher's exact test [9]. The null hypothesis ($H_0$) is that there is no association between the fact that the OS provides PI (variable PI) and the awareness of the use of resources (dependent variable). Conversely, the alternative hypothesis ($H_1$) is that the presence of PI helps Android users in identifying resource usage. We reject the null hypothesis if the p-value is lower than or equal to 0.05.

We run such an analysis in several scenarios:

- We considered only the tasks where the CB was latent (T1 and T4), to understand if the PI helps in possible malicious usage of the resources.
- We consider only the tasks where the use of the resources is explicit (T2 and T3), to understand if the PI still helps when the user probably already knows that the resource is used (*e.g.,* through the preview of the camera).

It is worth noting that there is no way participants in Group 1 and Group 2 could notice resource usage in Task 4 and Task 1 since they were *latent* and the PI were hiden. Thus, if such participants reported that they identified the resource usage, we excluded them from our study since this was a sign that they did not perform the task with sufficient attention, they provided random answers, or they could have not understood what they were asked to do.

In addition, we analyzed to what extent the resource (variable R) influenced the identification of PI. Since we run multiple comparisons, we use Benjamini & Hochberg method [10] to adjust the p-values.

Finally, we report in how many cases the correctness achieved in the *matching phase* (in which we showed pictures of the tasks performed with the PI visible) is higher than the one achieved after simply using the app. If after this phase the percentage improves, we can conclude that PI can capture the users' attention, but the focus on a particular task can make them miss the visual stimulus.

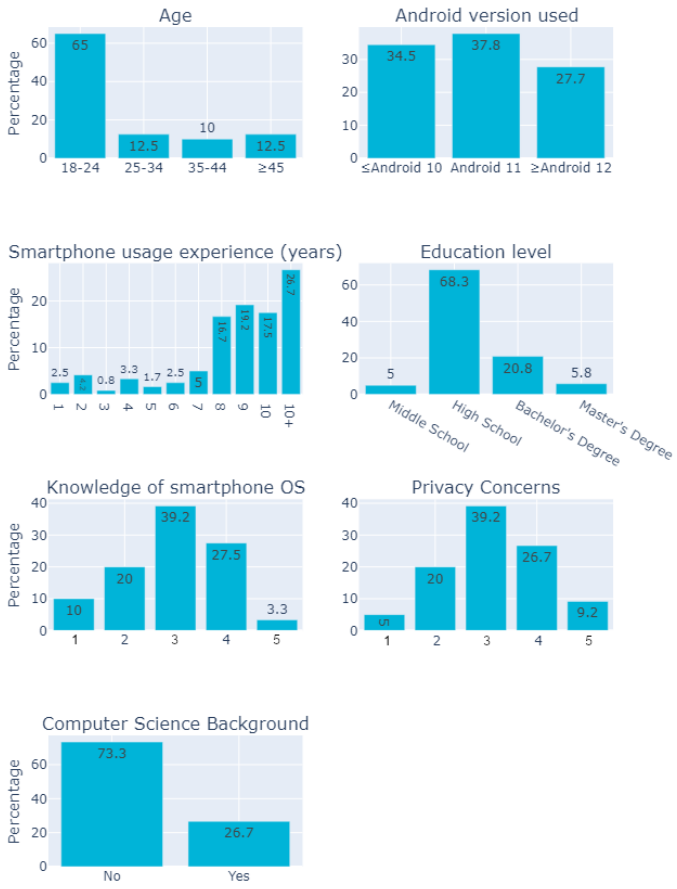



Fig. 4. Demographic users information.

### 3.4 Data availability

The data that support the findings of this study are openly available at https://dibt.unimol.it/report/privacy-indicators/, derived from users' responses to questionnaires available at https://dibt.unimol.it/report/privacy-indicators/#survey. Simulations of the four tasks can be observed at https://dibt.unimol.it/report/privacy-indicators/#tasks, with explanation of their behavior and visible PI. In addition, we have also made available prototypes that can be used directly on personal smartphones; they can be accessed by scanning the QR codes at https://dibt.unimol.it/report/privacy-indicators/#prototypes.

## 4 Empirical Study Results

Figure 4 summarizes the demographics of the sample we involved in our study. Since, as previously reported, we randomly assigned each participant to one of the two groups, there is a small difference in terms of number of participants in them (63 in Group 1 and 57 in Group 2). The two groups are generally balanced in terms of all the demographics we acquired. A quarter of the participants reported experience (education or job) in Computer Science (17 in Group 1 and 15 in Group 2). Concerning the age, we can observe that the majority of participants (∼65%) are in the range 18-24 years old (78, *i.e.*, 43 in Group 1 and 35 in Group 2), while we have ∼12.5% participants in the range 25-35 (15, *i.e.*, 7 in Group 1 and 8 in Group 2), ∼10% in the range 35-44 (12, *i.e.*, 4 in Group 1 and 8 in Group 2), and 12.5% that are 45 years old or more (*i.e.*, 9 in Group 1 and 6 in Group 2). Both groups have similar proportions of participants with a middle school education. However, Group 1 has a slightly higher proportion of participants with a high school diploma (73%) than Group 2 (63.2%), while Group 2 has a higher representation of participants with a Bachelor's degree (28.1%) compared to Group 1 (14.3%). Lastly, a small percentage of participants in both groups hold a master's degree. Figure 4 also shows descriptive data on smartphone usage in the general population. Approximately most of those respondents used smartphones for more than eight years.

Regarding the Android versions used, Android 12 or higher versions are used by 26.9% of Group 1 and 28.1% of Group 2, while Android 11 is used by 36.5% and 38.6% of Groups 1 and 2, respectively. Finally, 36.5% of Group 1 and 33.3% of Group 2 use Android versions earlier than 11. In other words, a significant portion of our sample daily use Android versions equipped with Privacy Indicators (Android 12 or later). Concerning privacy, low concern levels were rare in both groups, with just 1.6% of Group 1 and 8.8% of Group 2 selecting the minimum score. Most participants expressed moderate privacy concerns, with the majority in both groups selecting 3 on the scale. Despite these minor variations, both groups shared similar profiles in terms of smartphone knowledge and privacy awareness, though slight differences were apparent at the scale's extremes.

Applying the exclusion criterion previously discussed, we identified and excluded 2 participants from the final analysis. These individuals reported identifying latent resource use in tasks where privacy indicators were invisible by design.

### 4.1 Results Analysis

We analyze the correlation of factors in explicit and latent contexts separately. In Table 2 we report the percentage of participants who correctly identified the use of a resource in *latent* and *explicit* tasks, when PI are visible and hidden. Table 3 reports the same results divided by the type of resource involved (camera or microphone). As for the *explicit* context, we observe that the rate of correct identifications of the resource usage is only slightly higher when PI are visible (100 vs. 93). In this case, there is not a statistically significant association between resource identification and the use of PI (p-value = 0.3291). This means that we can not conclude that PI are useful for signaling the resource usage when the usage is explicit. When computing the effect size, we observe that the contribution of PI is *very small* (odds-ratio = 1.449) [11].

Something generally similar happens in the *latent* context, in which the rate of correct identifications of the resource usage is, again, slightly higher when PI are visible (6 vs. 0). For the *latent* context, the observed effect of PI is *marginally* significant (p = 0.0586). The odds-ratio is not a useful measure of effect size in this case: the lack of identification of resource usage when the PI is not visible is 0 by construction since there is no way participants could notice the resource usage, and this leads to infinite odds-ratio even when a single individual correctly identifies the

TABLE 2
Difference of resource identification between latent and explicit tasks in presence or absence of PI.

| % Identified | PI visible | PI not visible |
|---|---|---|
| *Latent* | 5% | 0% |
| *Explicit* | 83.3% | 77.5% |

TABLE 3
Summary table with the number of participants who correctly identified the use of the resource (camera or microphone) reported in the single row and the respective behavior (latent or explicit), when PI were visible and not visible.

| Context | Resource | *PI visible* | *PI non visible* | Task |
|---|---|---|---|---|
| ***Explicit*** | ***Camera*** | 53/57 *(92.9%)* | 58/63 *(92.1%)* | **2** |
| | ***Microphone*** | 47/63 *(74.6%)* | 35/57 *(61.4%)* | **3** |
| ***Latent*** | ***Camera*** | 4/63 *(6.3%)* | 0/57 *(0%)* | **1** |
| | ***Microphone*** | 2/57 *(3.5%)* | 0/63 *(0%)* | **4** |

resources. However, only 5% of the participants were able to identify the resource usage in this scenario. This means that the PI would not be effective for the large majority of Android users.

Table 3 highlights differences in the usage identification rate for camera and microphone. To analyze this phenomenon more in-depth, we checked if the type of resource used leads to a significant difference in the identification, regardless of the context or the presence of PI. In this case, we obtained a p-value = 0.0054 using the Fisher's test, indicating that the difference is, indeed, significant. In particular, the participants identified more easily the use of the camera than the use of the microphone. This behavior is evident in all tasks, in both latent and explicit context, with PI visible and hidden, except for tasks in which the behavior was latent and PI were not visible (where it is impossible to identify a resource). In the explicit context and with PI visible, an 18.3 percentage points difference can be observed between camera identification compared to the microphone identification. A higher difference (30.7 percentage points) is noticeable for hidden PI in the explicit context, where the camera usage is identified more frequently. A difference of 2.8 percentage points can be noted in the latent context with visible PI.

#### 4.1.1 Matching Phase

This section presents the analysis results conducted on the responses provided during the *matching phase*, taking into account only those participants who had not identified the resource used during the *response acquisition phase*. In Task 1, only 6 out of 59 participants (that previously had not identified the use of the resource) changed their minds by correctly selecting the screen related to the resource being used, even when the PI was visible, 2 of which are Android 12 users. Concerning Task 2, 2 out of 4 participants correctly selected the screen when the PI was shown. In Task 3, 2 out of 16 participants responded by correctly identifying the screen with the PI, and none of them used Android 12. In Task 4, 4 out of 55 participants correctly selected the screen with the PI, one of them is an Android 12 user. Table 4 summarizes the results of this analysis.

In conclusion, concerning the **RQ1** our results suggest that PI play a marginal role in enabling users to identify resource use.

TABLE 4
The number of participants that, during the *response acquisition phase*, did not identify the resource and instead, during the *matching phase*, changed their response correctly.

| Tasks | *# identified in the matching phase* |
|---|---|
| *1* | 6/59 |
| *2* | 2/4 |
| *3* | 2/16 |
| *4* | 4/55 |

### 4.2 Discussion

The goal of our study is to understand to what extent PI enhances users' awareness of the use of privacy-related resources in different contexts. Accordingly, it was critical to understand if users actually notice that a resource was used because of the PI.

The analysis performed allowed us to answer RQ1. The main goal of PI is to inform the user whenever an application makes use of a runtime permission to access a sensitive resource, like the microphone or the camera. Regarding this, we noticed that the contribution of PI to the identification of resources usage is not statistically significant. As a matter of fact, in spite of an increment in the total number of correctly identified cases, PI do not adequately enhance the users' awareness of when and where an application accesses a specific sensitive resource.

Concerning the tasks in which the access to the sensitive resource is explicit, we can notice a high identification rate independently of the PI. This is likely due to the fact that users have other elements to understand that the resource was accessed (*e.g.,* the camera preview on the screen). Considering the tasks in which camera was used, more than 90% of the users are able to identify when the application accessed the camera just by means of contextual information, *i.e.,* the task they are performing and what the application showed on the screen. It is worth noting that in a few cases (4 out of 57 participants with PI enabled and 5 out of 63 participants without PI enabled) the subjects did not correctly identify the use of the camera, despite the onscreen preview. This is probably due to the fact that, during the experimentation, the preview showed a default animation, since we used a simulation of the OS instead of the real OS. This could have misled some of the subjects. Overall, the difference between the results in which the PI were visible and those in which they were not visible is negligible (0.8 percentage points improvement on camera).

Considering the task where the application made an explicit use of the microphone, the difference between the two treatments is higher. In fact, when PI were visible 47 out of 63 participants (about 74.6%) were able to identify the fact that the microphone was used, as compared to 35 out of 57 participants (about 61.4%) that noticed the use of microphone when PI were not visible. It is worth noting that, in this scenario, the identification rate is lower in both cases. This is likely due to the fact that the visual feedback of the microphone enabling (during the voice call, the icon to turn the microphone off and on is visible) is less evident

than the camera preview (even if the preview did not show the real camera, but a predefined video).

Even if we did not find a statistically significant correlation between the presence or absence of the PI and the identification of the use of the microphone, we noted an improvement of more than 13 percentage points in the identification when PI were visible. Overall, our results suggest that the PI enhance the user awareness of the resource usage, but only very slightly.

To further assess this thesis we discuss what happens when the use of the same kind of resource is made within a latent context, *i.e.,* when there is not a clear reason to access the resource and there is no visual feedback to the user to inform that the app is accessing it, except for the PI, when shown. In particular, we removed any information that could help them to understand the use of the resource, *i.e.,* the camera preview and the behavior that was related to the microphone activation. We expect that this scenario is more adequate to test the effectiveness of the PI, as the participants cannot guess whether the app is using the camera or the microphone by reasoning about the task or noticing the resource activation, but can be informed by the system only by means of the PI. It is worth noting that this is not an unreal scenario as it is likely what happens when the user is victim of a spyware and other type of malware applications. This is exactly where we expect the PI helps the user identify the malicious behaviour.

By construction, none of the participants not having the PI feature enabled was able to notice that the camera and the microphone have been used during the tasks. However, even when PI were visible, only 4 out of 63 participants (6.3%) correctly identified the use of the camera while 59 (93.7%) did not. Similarly, 2 out of 57 participants (3.5%) correctly identified when the microphone was used, while 55 (96.5%) failed. Note that, in this case, the contribution of PI to the awareness of resource usage is even lower than the result achieved in the explicit context. This is counter-intuitive, as we expected at least the same improvement.

To conclude, despite an improvement can be noted, it is clear that PI, in their current implementation, are still unsatisfactory as a solution to make users aware of the resources accessed during the use of Android apps.

It is worth noting that most of the participants do not use Android 12 or later as their primary mobile OS, and that such a feature probably requires time to be effectively mastered by the users. Thus, we have conducted a further analysis restricting the observations to Android 12 or later users only, to confirm or contradict our findings. Furthermore, we are interested only in tasks where the context was latent, since we noted that when the context is explicit users are able to identify the use of sensitive resources in any case. Among the 33 participants that claimed to use regularly Android 12 and later, 17 were involved in Task 1 and 16 in Task 4. Only 3 of the participants involved in Task 1 and 1 participant involved in Task 4 noticed that the camera and the microphone have been used during the tasks. While this result is slightly better, the message is the same: the large majority of participants (87%) fail to identify the use of sensitive resources.

It is possible to observe that the results are not more encouraging after the *matching phase*. During this phase, the user is not distracted by the execution of the task and statically observes screens related to already executed tasks. Note that the screens show the PI when enabled. Concerning the tasks where the use of resources was explicit, the improvement for the camera task was 3.5 percentage points (2 users of the 4 who initially did not identify the camera noticed the PI), while for the microphone task the improvement was 3.2 percentage points (2 users out of 16). Note that the results were quite good for explicit contexts, thus we expected a limited improvement during the *matching phase*. As expected, the most noticeable improvement was in the tasks making a latent use of resources. In particular, concerning the camera task it is possible to observe an improvement of 9.5 percentage points (6 users out of 59 who initially did not identify the camera), while in the microphone task the improvement was 7 percentage points (4 users out of 55). This suggests that when the PI is shown statically, participants are substantially more able to identify the resource usage. Unfortunately, when the use of a resource is latent, the user awareness is still very limited. In fact, when the PI was visible, 53 users did not notice that the camera was used during the task (51 for the microphone).

We deliberately concentrated on evaluating the use of the camera and microphone, as these are currently the only resources for which Android PI are used. By doing so, we ensured that our experiment design mirrored real-world scenarios, allowing us to assess the effectiveness of PI under conditions that Android users regularly encounter. Our results in a latent scenario, where the resource use isn't immediately apparent, could potentially apply to other resources that don't provide clear visual feedback when accessed, such as contacts or data storage. However, we consider this a preliminary interpretation and suggest that further research is necessary to conclusively extend our findings to these resources.

Moreover, we conducted an analysis to investigate whether the awareness of resource usage was affected by the characteristics of the participants we involved (such as their educational level or how they feel about privacy). We focused on analyzing how people behave when presented with PI in latent tasks where the PI is the only mean through which participants could notice the resource usage. To achieve this goal, we used logistic regression. In our case, the independent variables are all demographics, while the dependent variable is the identification of the resource usage. We found that none of the characteristics we looked at seemed to have a significant effect on whether participants noticed the resource usage. This means that we did not find any evidence to suggest that things like how long someone has been using a smartphone or whether they have a background in Computer Science would make them more or less likely to notice the PI. For example, our results suggest that subjects with a high level of education is not more likely to notice the PI than subjects with a lower level of education. Similarly, having a background in Computer Science did not increase the chances of noticing the PI compared to not having a computer science background. The absence of significant predictors suggests that PI awareness, and thereby the latent behavior associated with the usage of the resource it represents, might be intrinsic to how it is implemented.

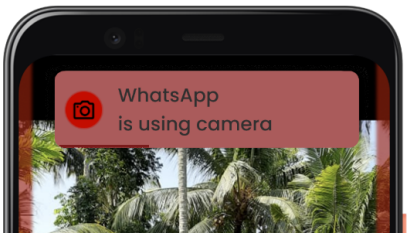



Fig. 5. An example of our enhanced privacy indicator.

## 5 Evaluating an Alternative Solution

There are many reasons why PI may not be effective in supporting awareness of the use of privacy-related resources. Two of the contributing factors could be the visibility of the indicators and the association between PI and the fact that a sensitive resource is used. In order to evaluate whether this conjecture is correct and the user awareness can be enhanced, we introduce an alternative version of PI which is clearly more visible and that highlighted the resource accessed by the app. Our solution (named POPUP) is a non-blocking and informative notification implemented through a toast component similar to push notifications.

The POPUP has three visual enhancements with respect to standard PI (see Figure 5):

- **Icon**: a red icon representing the resource accessed (e.g., a camera or microphone icon).
- **Description**: a textual information indicating the application that is accessing the resource.
- **Animation**: the POPUP includes a red progress bar that represents how long the textual notification will be visible. Moreover, POPUP includes two red bands appearing on the screen's sides. These bands alternate between widening and narrowing, providing a peripheral alert that may capture users' attention more effectively even when the textual notification is hidden.

The POPUP was designed to have a limited impact on the user experience with the application. Indeed, it is non-blocking (similarly to the default Android implementation of PI) and disappears automatically after 5 seconds so that the user can ignore it. Nonetheless, it provides enough cues to make the user aware of the resource accessed by the app.

### 5.1 Validation of POPUP

To validate POPUP, we run an empirical study very similar to the previously presented one.

#### 5.1.1 Empirical Study Design

The goal was to understand if our solution is more effective than the default one provided, at the moment, in Android. Such a study is guided by the following research question:

**RQ2** *Does an enhanced indicator improve user ability to identifying when a resource is used by a mobile application compared to default Android PI?*

We entirely re-used the design of the previous experiment (*e.g.,* the apps and the tasks). In this case, we involved

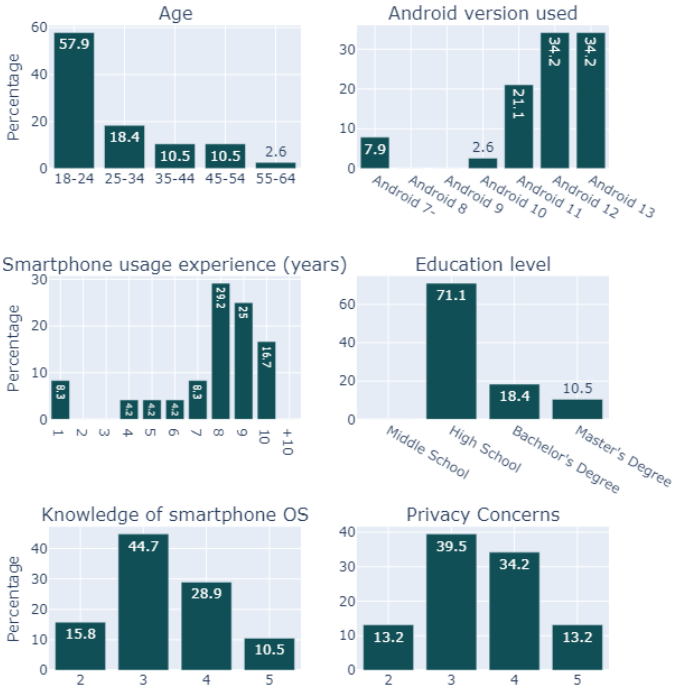



Fig. 6. Demographic users information in our solution

**38 Android users**. Some of them participated in the first experiment and expressed an interest in being involved again in this study, while others were volunteers from a public event. The two treatments, in this case, were (i) default PI, and (ii) POPUP PI. Since all the participants were exposed to both default PI and POPUP PI (in different tasks), we needed to re-think the instructional tutorial presented to the participants. In the previous experiment, it only presented default PI, while in this case, we wanted to present our new solution as well. Thus, we presented both of them, explicitly reporting that they could encounter either and that the meaning was exactly the same.

As for data analysis, we employed a Fisher's exact test, with the null hypothesis ($H_0$) that ***"Enhanced indicators do not significantly improve user ability to identifying when a resource is used by a mobile application compared to default Android PI,"*** and the alternative hypothesis ($H_1$) that ***"Enhanced indicators significantly improve user ability to identifying when a resource is used by a mobile application compared to standard PI."***

#### 5.1.2 Empirical Study Results

Figure 6 visualizes the demographic distribution of the participants, while we present in Table 5 the results of our experiment.

For the *explicit* context, the obtained p-value is **0.0125**, a value considerably below the 0.05 threshold. This suggests a significant correlation between the use of enhanced indicators and higher user awareness of resource usage. Note that, in this case, statistical significance has been achieved with a much smaller number of participants, thus suggesting a larger effect of the treatment. Indeed, the odds-ratio is 7.194 (*i.e., large* [11]), suggesting that users are much more likely to

TABLE 5
Summary of the experiment results

| Context | Resource | *PI visible* | *POPUP* | Task |
|---|---|---|---|---|
| ***Explicit*** | ***Camera*** | 15/19 *(78.95%)* | 19/19 *(100%)* | **2** |
| | ***Microphone*** | 13/19 *(68.42%)* | 17/19 *(89.47%)* | **3** |
| ***Latent*** | ***Camera*** | 0/19 *(0%)* | 16/19 *(84.21%)* | **1** |
| | ***Microphone*** | 4/19 *(21.05%)* | 11/19 *(58.89%)* | **4** |

recognize the resource usage when enhanced indicators are used compared to default Android PI. This result is even more clear in the *latent* context, for which the p-value is extremely low (**1.939e-07**) and the odds-ratio, again, *large* (20).

These results clearly show that a more prominent indication of resource usage significantly helps users.

## 6 THREATS TO VALIDITY

Threats to *internal validity* concern confounding factors that could influence the results. Participants were randomly assigned to one of the two groups, countering selection bias by making groups comparable at the start of the study. We assigned each participant 4 tasks; it is reasonable to think they get used to complete them by noticing how previous tasks work, increasing the learning effect. We changed the treatment between tasks assigned to the participants (based on the variables reported in Section 3.2) to reduce the learning effect. To reduce the risk of possible random responses from participants we included control questions to sustain participant focus and disqualified individuals who inaccurately identified latent resource usage when PI were not visible. Only two participants out of the initial 122, less than 2%, were excluded based on this criterion. This minimal exclusion rate does not significantly compromise the overall validity and reliability of our experiment. The total percentage of correct responses to control questions across all participants turned out to be **94.17%**. This high percentage suggests that participants generally demonstrated a strong understanding of the tasks and were sufficiently focused.

One of the possible risks is that the results depend on the specific task in two groups and they may not be balanced since it is possible to observe differences in the results of individual tasks. For that reason, we performed an analysis with Fisher's test based on each task within both groups. Results ruled out this possibility for all tasks ($p_{Task1}$ = 0.30, $p_{Task2}$ = 1, $p_{Task3}$ = 0.30, $p_{Task4}$ = 0.30).

In Section 4.2 we discuss about the improvement given by PI to participants' awareness of resource usage between latent and explicit tasks, noting that the improvement is lower in latent tasks than in explicit tasks. This result could be counter-intuitive and could indicate that the aforementioned results are not strictly related to PI. We mitigated this thread through the matching phase where it is possible to observe that the results do not improve significantly.

To detect errors and improve the experimental design, we conducted a pilot study with 27 participants. To avoid as many problems as possible related to social interaction, we have selected pilot study participants outside the university setting to minimize possibles contacts between them and actual study participants. In addition, to limit interactions among participants within the university, during the final study participants were grouped in sessions and monitored by university staff. Moreover, we prevented, as much as possible, any contact between participants from different sessions.

Another threat to validity is that 72.3% of users do not run Android 12 or later versions on their own devices, so they may not be aware of PI. For this reason, at the beginning of the experiment, we explained the new privacy-related features introduced in Android 12, with particular reference to PI. In addition, we created a prototype where participants were shown a tour of the new features, interactively demonstrating how PI works. The tour was mandatory and was rerun in case the participants did not declare they fully understood the new feature introduced in Android 12. However, the risk that more experienced users on the latest version of Android would obtain different results exists. To this aim, we have conducted a further analysis restricting the study to regular Android 12 and later users only. Even in this case, 87% of them failed to identify the use of sensitive resources in latent contexts. It is important to note that this is a qualitative analysis conducted on a limited sample of the participants, so it is recommended that a replication of this study will be conducted in the future involving subjects with a higher experience on Android 12 and later to confirm or contradict our findings.

We decided to limit the interactions that users can do on the prototypes; in this way, it is tough to get stuck during the steps to be performed, informing participants during the tutorial that by doing an action that is not allowed, the prototype will highlight with a blue container any of the allowed interaction. Despite this is not the actual usage scenario, we believe that the restricted interaction do not significantly impact on the user's understanding of whether the app uses the camera or the microphone or not.

We did not explicitly disclose the goal of the study to the participants to avoid creating bias, but the purpose could be evident to our participants at various moments, which may have influenced their conduct. For example, before each task, we notified the user that specific task represented a smartphone running Android 12 and we clarified that all permissions were accepted by default and would not be asked again. In this way, we might have suggested that our goal was related to resource usage, but this was necessary because the PI is only available since Android 12. Moreover, we wanted to make participants aware of the fact that no popup request for permission would be presented during the tasks. Note that making any request explicit would have biased the results of the study. In fact, within a task having a latent access to the microphone, the presence of the popup asking for permission to use it would have made the task explicit, making the PI useless. We wanted to focus the participants' attention only on the PI, avoiding other identifiers of resource use; moreover, the request to use a permission can occur only once during app use (unless the user agrees only for this one time), so we did not affect normal behavior.

Threats to *external validity* concern the generalization of the experiment and our findings. Our sample has the risk of not being representative of the entire population of Android users, as the study was conducted in a University

setting. However, the participants have different characteristics from each other, as identified during the demographic questionnaire.

There are characteristics of our study design, like any empirical investigation, that may restrict the generalizability of the results. Our focus on popular and trusted apps assumes that users are more likely to grant them full access to sensitive resources. However, this might not hold true for lesser-known or non-trusted apps, where users may be more cautious. Therefore, our findings might not generalize effectively to these scenarios, warranting further research.

We performed our experiment in a highly standardized and somewhat artificial environment, and our findings could not reflect user behavior in actual apps usage. We followed recommendations for experimental studies proposed by Aguinis in [12] to reduce this problem. For example, our participants were given a thorough explanation (both orally and in writing) of what they were supposed to do. The individual goals to be achieved in the tasks were clearly explained. In addition, they were given a tutorial on how the prototypes to use work. All participants were able to complete the tutorial, obtaining the check-code to enter in order to proceed (as in the individual tasks) without any problems. These measures, combined with the use of their own smartphone and realistic prototypes experiencing interface and action to perform identically to real apps, ensure a high level of immersion for the participants, which, as previous work has shown, leads to the highest possible generalizability of the study results [12] [13] [14].

## 7 Related Work

Security and privacy are mobile applications primary concerns, strongly related to the Android permission model and user behavior. Several works [6] have emphasized user behavior when granting app permission. However, although there have been many improvements to the authorization system over time, mainly aimed at increasing the perceived level of security and increasing control over how and when applications access personal data, it has not yet been possible to achieve adequate levels of user awareness and flexibility [15]. People often grant permission without paying much attention because they think the app would not work otherwise, ignoring the possible impact on their privacy. Many users ignore operating system warnings about permission requests [16] [17] [6] [18], and grant the app permission that may disclose sensitive data.

Different elements of mobile applications have been studied, and solutions have been proposed. The main weakness of the model currently in use is represented by the fact that the approval of a single request is automatically extended to the entire application life-cycle. Once a permission is granted to the app, it has the possibility to access that particular type of data or critical resources, without the need for further consent and potentially even in the background [2]. Because of this substantial limitation, the various approaches proposed over time have emphasized two main characteristics: the contextualization of requests, which puts the user in a position to make a more informed choice, postponing the final decision to the moment when the application actually accesses it [19], the level of granularity of the controls, which thanks to a wider choice of combinations allows to capture with an increasingly better approximation the user's point of view on the management of his digital life [15]. The latter can be achieved not only by increasing the kinds of permissions, but also by providing different degrees of detail, or different behaviors depending on the particular context of use.

Shen et al. [8] investigated the problem through the analysis of real users permission settings and large-scale user studies. They found that users have several common misunderstandings on specific permissions usage, and many Android users are not aware of permission model changes. However, their goal was to assess how well users identify the scope of permissions to obtain information that systems can provide to help users make more informed permission decisions. Instead, our work evaluates the extent to which users are helped by PI in identifying the use of a resource.

Elbitar et al. [20] investigated the effects of timing and rationales on users runtime permission decisions and showed that they affect users permission decisions and the evaluation of their decisions. They found that the effect of timing and rationale depend on one another and should not be evaluated separately. Based on the achieved results, they suggest that the current Google guidelines should be refined to better aid users in their decision-making process. They suggest that current mobile platforms could benefit from a customized solution on a per-user basis, in which users can define when permissions should be requested and whether rationales should be given. However, in their work they considered the effects of factors (timing/rationales) different from our (PI). We evaluate the effectiveness of PI that are not dependent on developers' actions as in the case of timing or rationales. We focused on an Android feature rather than developer practices, even though both have an effect on user awareness on permission usage.

Scoccia et al. [21] conducted a study to investigate how end users perceive the runtime permission system of Android, inspecting user reviews on apps published in the Google Play Store. They suggest that permission-related issues are widespread and determined recurring points made by users about the new permission system and classified them into a taxonomy. However, in their work they considered changes to the permission model on Android 6, and PI had not yet been introduced. In addition, they acquire data from reviews on the Play Store with the goal of creating a taxonomy of recurring points made by users about Android 6 permission system, we, on the other hand, wanted to evaluate the effectiveness of the permission system, through statistical analysis, from Android 12 by conducting a controlled experiment directly with users.

## 8 Conclusion and Future Works

In this paper, we conducted a preliminary study on the effectiveness of the PI feature released in Android since version 12 to inform the user whenever an app accesses the camera or the microphone of the mobile device.

We conducted a controlled experiment involving 122 participants, who were asked to complete a series of tasks on a series of prototypes that mimic Android 12 with the PI

feature enabled and disabled. Some of the tasks explicitly made use of the camera or microphone, while other tasks used them in a latent manner. We wanted to assess how effective are PI to help users identifying the use of a resource (camera or microphone) in different scenarios.

Our results show no significant correlation between the presence or absence of the PI and the detection of the resource used, suggesting that most of Android users do not adequately notice their presence. Interestingly, our results are independent of factors like privacy perception level, knowledge of smartphone operating systems, years of smartphone usage, level of education, and having a Computer Science background.

We, therefore, evaluated an alternative solution focused on two of the aspects that we believe could contribute to a reduced effectiveness of the PI, namely their visibility and their ability to adequately inform the user that the app has accessed a specific resource. The results of this second study demonstrated that, by acting on the way PI are implemented, a significant improvement in their effectiveness can be achieved.

Our work was focused only on two specific privacy-related resources, although we can imagine that Android will use a similar approach for other runtime permissions in the future. We suppose that it will be even less effective extending the approach to include other runtime permissions, (*e.g.*, ACCESS_FINE_LOCATION, INTERNET_CONNECTION, WRITE/READ_EXTERNAL_STORAGE) since these are even less explicit in their use. Moreover, the risk exists that the user could be further confused by several indicators repeatedly shown in the toolbar and would barely remember their meaning or even stop paying enough attention to the LED displayed on top of the screen. On the other hand, focusing only on these two resources is limiting since, although they are among the most sensitive permissions, many other threats to the privacy are related to different runtime permissions. Therefore, part of our future agenda will include more in-depth studies to understand whether users tend to identify access to other sensitive resources and whether we can further generalize the obtained results.

## REFERENCES


[1] M. Guerra, R. Milanese, R. Oliveto, and F. Fasano, "RPCDroid: Runtime Identification of Permission Usage Contexts in Android Applications," in *Proceedings of the 9th International Conference on Information Systems Security and Privacy, ICISSP 2023, Lisbon, Portugal, February 22-24, 2023*, P. Mori, G. Lenzini, and S. Furnell, Eds. SciTePress, 2023, pp. 714–721. [Online]. Available: https://doi.org/10.5220/0011797200003405

[2] P. Wijesekera, A. Baokar, L. Tsai, J. Reardon, S. Egelman, D. Wagner, and K. Beznosov, "Dynamically regulating mobile application permissions," *IEEE Security Privacy*, vol. 16, no. 1, pp. 64–71, 2018.

[3] M. A. Cohen, P. Cavanagh, M. M. Chun, and K. Nakayama, "The attentional requirements of consciousness," *Trends in Cognitive Sciences*, vol. 16, no. 8, pp. 411–417, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364661312001519

[4] A. CHALMERS and K. CATER, "41 - exploiting human visual perception in visualization," in *Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds. Burlington: Butterworth-Heinemann, 2005, pp. 807–816. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123875822500435

[5] A. Mack, "Inattentional blindness: Looking without seeing," *Current Directions in Psychological Science*, vol. 12, no. 5, pp. 180–184, 2003. [Online]. Available: https://doi.org/10.1111/1467-8721.01256

[6] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*, ser. SOUPS '12. New York, NY, USA: Association for Computing Machinery, 2012. [Online]. Available: https://doi.org/10.1145/2335356.2335360

[7] A. Peruma, J. Palmerino, and D. E. Krutz, "Investigating user perception and comprehension of android permission models," in *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*, ser. MOBILESoft '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 56–66. [Online]. Available: https://doi.org/10.1145/3197231.3197246

[8] B. Shen, L. Wei, C. Xiang, Y. Wu, M. Shen, Y. Zhou, and X. Jin, "Can systems explain permissions better? understanding users' misperceptions under smartphone runtime permission model," in *USENIX Security Symposium*, 2021.

[9] R. A. Fisher, "Statistical methods for research workers," in *Breakthroughs in statistics*. Springer, 1992, pp. 66–70.

[10] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995. [Online]. Available: http://www.jstor.org/stable/2346101

[11] H. Chen, P. Cohen, and S. Chen, "How big is a big odds ratio? interpreting the magnitudes of odds ratios in epidemiological studies," *Communications in Statistics—simulation and Computation®*, vol. 39, no. 4, pp. 860–864, 2010.

[12] H. Aguinis and K. J. Bradley, "Best practice recommendations for designing and implementing experimental vignette methodology studies," *Organizational Research Methods*, vol. 17, no. 4, pp. 351–371, 2014. [Online]. Available: https://doi.org/10.1177/1094428114547952

[13] D. J. Woehr and C. E. Lance, "Paper people versus direct observation: An empirical examination of laboratory methodologies," *Journal of Organizational Behavior*, vol. 12, pp. 387–397, 1991.

[14] R. Hughes and M. Huby, "The application of vignettes in social and nursing research," *Journal of advanced nursing*, vol. 37, pp. 382–6, 03 2002.

[15] G. L. Scoccia, I. Malavolta, M. Autili, A. Di Salle, and P. Inverardi, "Enhancing trustability of android applications via user-centric flexible permissions," *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2032–2051, 2021.

[16] D. G. N. Benítez-Mejía, G. Sánchez-Pérez, and L. K. Toscano-Medina, "Android applications and security breach," in *Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)*, 2016, pp. 164–169.

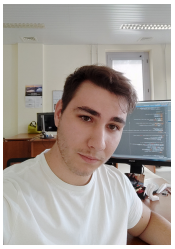
[17] M. Benisch, P. Kelley, N. Sadeh, and L. Cranor, "Capturing location-privacy preferences: Quantifying accuracy and user-burden tradeoffs," *Personal and Ubiquitous Computing*, vol. 15, pp. 679–694, 10 2011.

[18] P. Kelly, S. Consolvo, L. Cranor, J. Jung, N. Sadeh, and D. Wetherall, "A conundrum of permissions: Installing applications on an android smartphone," in *Proc. of USEC 2012*, vol. 7398, 03 2012.

[19] P. Wijesekera, A. Baokar, L. Tsai, J. Reardon, S. Egelman, D. Wagner, and K. Beznosov, "The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences," in *2017 IEEE Symposium on Security and Privacy*, 2017, pp. 1077–1093.

[20] Y. Elbitar, M. Schilling, T. T. Nguyen, M. Backes, and S. Bugiel, "Explanation beats context: The effect of timing & rationales on users' runtime permission decisions," in *USENIX Security Symposium*, 2021.

[21] G. L. Scoccia, S. Ruberto, I. Malavolta, M. Autili, and P. Inverardi, "An investigation into android run-time permissions from the end users' perspective," in *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*, ser. MOBILESoft '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 45–55. [Online]. Available: https://doi.org/10.1145/3197231.3197236



**Michele Guerra** He earned his Bachelor's and Master's degrees in Computer Science from the University of Molise, Italy, and has been a Ph.D. student there since 2021, guided by Professors Rocco Oliveto and Fausto Fasano. His research is deeply rooted in Software Engineering, with a special focus on Android Security and Privacy, Automated Testing, Software Quality, and Empirical Software Engineering.

**Simone Scalabrino** Simone Scalabrino is a research fellow and adjunct professor at the University of Molise. His research interests are related to Software Engineering and, specifically, to the internal quality of software systems (*e.g.,* understandability and readability of code), testing (*e.g.,* automatic generation of test cases) and security (*e.g.,* identification of vulnerabilities in software systems).

**Fausto Fasano** is Associate Professor in the Department of Bioscience and Territory at University of Molise (Italy). He is the Vice-Chair of the Computer Science program and Director of the MOSAIC Research Center of the University of Molise. He received the PhD in Computer Science from University of Salerno (Italy) in 2007. His research interests include mobile applications security, global software engineering, software maintenance and evolution, and empirical software engineering.

**Rocco Oliveto** is a Professor in the Department of Bioscience and Territory at University of Molise (Italy). He is the Chair of the Computer Science program and the Director of the Laboratory of Computer Science and Scientific Computation of the University of Molise. He received the PhD in Computer Science from University of Salerno (Italy) in 2008. His research interests include traceability management, information retrieval, software maintenance and evolution, search-based software engineering, and empirical software engineering. He is author of about 150 papers appeared in international journals, conferences and workshops. He serves and has served as organizing and program committee member of international conferences in the field of software engineering. He is a member of IEEE Computer Society and ACM.