Evaluation of Personalized Security Indicators as an Anti-Phishing Mechanism for Smartphone Applications

Claudio Marforio[1], **Ramya Jayaram Masti**[1], **Claudio Soriente**[2], **Kari Kostiainen**[1], **Srdjan Čapkun**[1]

[1]Institute of Information Security, ETH Zurich
Zürich, Switzerland
[claudio.marforio,ramya.masti,
kari.kostiainen,srdjan.capkun]@inf.ethz.ch

[2]Telefónica Research
Barcelona, Spain
claudio.soriente@telefonica.com

ABSTRACT

Mobile application phishing happens when a malicious mobile application masquerades as a legitimate one to steal user credentials. Personalized security indicators may help users to detect phishing attacks, but rely on the user's alertness. Previous studies in the context of website phishing have shown that users tend to ignore personalized security indicators and fall victim to attacks despite their deployment. Consequently, the research community has deemed personalized security indicators an ineffective phishing detection mechanism.

We revisit the question of personalized security indicator effectiveness and evaluate them in the previously unexplored and increasingly important context of mobile applications. We conducted a user study with 221 participants and found that the deployment of personalized security indicators decreased the phishing attack success rate to 50%. Personalized security indicators can, therefore, help phishing detection in mobile applications and their reputation as an anti-phishing mechanism in the mobile context should be reconsidered.

Author Keywords

Mobile Security; Phishing; Security Indicators.

ACM Classification Keywords

K.6.5. Security and Protection: Authentication

INTRODUCTION

Application phishing attacks in mobile platforms occur when malicious applications mimic the user interface (UI) of legitimate applications to steal user credentials. Phishing applications have been reported in the wild [14,34,42] with successful phishing attacks targeting thousands of users and procuring high revenues for the attackers [16]. Mobile phishing applications do not exploit system vulnerabilities [15]. They instead use standard system features and APIs, and leverage the user's incapacity to distinguish the legitimate application from a phishing one.

CHI'16, May 07 - 12, 2016, San Jose, CA, USA

ACM 978-1-4503-3362-7/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858085

Online services use *personalized security indicators* to aid the user in distinguishing the legitimate website from a phishing one [3,37]. The personalized security indicator (or "indicator" from now on) is an image chosen by the user when he enrolls for the online service. After enrollment, the website displays the indicator every time the user logs in. The indicator allows the user to authenticate the website and the user should enter his credentials only if the website displays the correct indicator.

Mobile applications can also use indicators to mitigate application phishing attacks [4,40]. The user chooses the indicator when he installs the application and must check that the application shows the correct indicator at each login. The indicator is stored by the application and the mobile OS prevents access from other applications.

In this paper, we start by categorizing application phishing attacks in mobile platforms and possible countermeasures. We show that all known countermeasures incur a tradeoff in security, usability and deployability. The benefits of security indicators are that they can counter many phishing attack vectors and implementation techniques, and they can be easily deployed by service providers since they do not require changes to the mobile platform or to the marketplace infrastructure.

Personalized indicators, however, rely on the user to detect phishing by checking the presence of the correct indicator. Previous work in the context of websites has shown that users tend to ignore personalized indicators when entering their login credentials [23,33]. We revisit the question of personalized indicator effectiveness and evaluate them in the previously unexplored context of smartphone applications.

Our rationale for evaluating indicators in this setting is that mobile user interfaces are considerably simpler than the ones of websites designed for PC platforms. As the user's focus is limited to a few visual elements, personalized indicators may be more salient in mobile application UIs [8,31]. Also, the usage patterns of mobile applications is different from those of websites, which may improve the detection of incorrect or missing UI elements. Additionally, the research community found browser security warning implementations ineffective [10, 12, 36],

but a recent study on newer implementations showed the opposite [1]. We argue that it is important to re-evaluate the effectiveness of security mechanisms when their implementations or deployment models have changed significantly.

Over one week, 221 study participants used a banking application we developed on their own smartphones to complete various e-banking tasks. On the last day of the study, we launched a phishing attack. Approximately 50% of the participants that used security indicators detected the attack and did not enter their credentials.

While further studies are still needed to gain more confidence in the effectiveness of personalized security indicators, this first study on smartphones shows that indicators can be more effective than previously believed when deployed in the mobile applications context.

To summarize, we make the following contributions:

- We analyze mobile application phishing attacks and possible countermeasures. We conclude that none of the countermeasures prevents all attacks and the problem of phishing remains largely unsolved.
- We report the results from a first user study that evaluates personalized indicators on smartphone applications. In our study, the deployment of indicators prevented half of the phishing attacks.
- We outline directions for further research that is needed to better assess the effectiveness of indicators as an anti-phishing mechanism under various deployment models.

PHISHING ATTACKS AND COUNTERMEASURES

In this section we categorize application phishing attacks on smartphones. All attacks are effective on Android and one of them also works for iOS. We discuss possible countermeasures and analyze them with respect to security, usability and deployment.

Phishing Attacks

Similarity attack

The phishing application has a name, icon, and UI that are similar or identical to the legitimate application. The adversary must induce the user to install the phishing application in place of the legitimate one. Successful similarity attacks have been reported for Android [11, 14, 16, 34] and iOS [25].

Forwarding attack

Another phishing technique is to exploit the application forwarding functionality of Android [15]. A malicious application prompts the user to share an event (e.g., a highscore in a game) on a social network and shows a button to start the social network application. When the user taps the button, the malicious application does not launch the social network application, but rather displays a phishing screen. The phishing screen asks the user to enter the credentials to access his account on the social network. Application forwarding is a common feature of Android and forwarding attacks may therefore be difficult for the user to detect.

Background attack

The phishing application waits in the background and uses the Android ActivityManager, or a side-channel [24], to monitor other running applications. When the user starts the legitimate application, the phishing application activates itself in the foreground and displays a phishing screen [4, 15].

Notification attack

The attacker shows a fake notification and asks the user to enter his credentials [40]. The notification window can be customized by the adversary to mimic the appearance of the legitimate application.

Floating attack

The attacker leverages the Android feature that allows one application to draw an Activity on top of the application in the foreground. This feature is used by applications to always keep a window in the foreground, for example, to display floating sticky notes. A phishing application that has the `SYSTEM_ALERT_WINDOW` permission can draw a transparent input field on top of the password input field of the legitimate application. The UI of the legitimate application remains visible to the user who has no means to detect the overlaid input field. When the user taps on the password field to enter his password, the focus is transferred to the phishing application which receives the password entered by the user.

Phishing Countermeasures

None of the attacks we discuss exploit OS vulnerabilities, but rather use standard Android features and APIs. Therefore, security mechanisms on the device (e.g., sandboxing or permission-based access control) or security screening run by the marketplace operator cannot prevent such attacks.

Similar to website phishing, thwarting application phishing attacks requires tailored security mechanisms. We describe possible countermeasures and categorize them in terms of security, usability and ease of deployment.

Signature-based detection

Signature-based malware detection techniques that look for patterns of system calls and permissions can be implemented by the marketplace operator (e.g., the Google Bouncer system [17]). Recently, the authors of [4] developed a static analysis tool to detect the use of APIs that enable background attacks. The drawback of signature-based detection solutions is that many phishing attacks (e.g., forwarding and similarity attacks) do not require specific API calls and would not be detected. This approach, therefore, applies only to a subset of possible attacks.

Name similarity

Marketplace operators can attempt to detect similarity attacks by searching for applications with similar names

	Marketplace Phishing Detection			On-device Phishing Prevention			
	Signature-based detection	**Name similarity**	**Visual similarity**	**Limited multi-tasking**	**Application name**	**Visual similarity**	**Personal indicator**
attacks							
similarity attack	–	+	–	–	–	+	–
forwarding attack	–	–	+	–	+	+	+
background attack	+	–	+	+	+	+	+
notification attack	+	–	–	+	+	–	+
floating attack	–	–	–	+	+	–	+
security							
false positives/negatives	–	–	–	+	+	–	+
reliance on user alertness	+	+	+	+	–	+	–
usability							
user effort at installation	+	+	+	+	+	+	–
user effort at runtime	+	+	+	+	–	+	–
restrictions on device functionality	+	+	+	–	–[1]	+	+
significant performance overhead	+	+	+	+	+	–	+
deployment							
changes to application provider (e.g., bank)	+	+	+	+	+	+	+
changes to marketplace	–	–[2]	–[2]	+	+	+	+
changes to mobile OS	+	+	+	–	–	–	+
changes to application	+	+	+	+	+	+	–

[1]restriction to full-screen applications with constant user interaction (Android Immersive mode)

[2]to check for phishing applications installed via sideloading

Table 1: Comparison of mechanisms to prevent application phishing attacks in mobile platforms. For each solution, a '+' represents a positive aspect, while a '–' a drawback.

or icons. Since many legitimate applications have similar names or icons (e.g., banking applications for the same bank in different countries), this approach would produce a significant number of false positives. Detecting phishing applications in the marketplace does not rely on the user's alertness or change the user experience. Checking for phishing applications installed from the web or from third-party marketplaces (sideloading) could leverage the Google App Verification service [18].

Visual similarity

The marketplace operator can attempt to mitigate background or forwarding attacks by searching for applications with similar UIs and, in particular, similar login screens. UI extraction and exploration are challenging problems and none of the known techniques provides full coverage [2]. Another option is to perform visual similarity comparisons directly on the device. In [26] the authors propose periodically taking screenshots and comparing them to the login screens of installed applications. While this solution does not incur the problem of UI extraction, it incurs a significant runtime overhead.

In general, if detection is based on matching UIs, phishing applications that use a slightly modified version of the legitimate application UI may go unnoticed. Finding an effective tradeoff (a similarity threshold) is a challenging task and is likely to include both false positives and negatives [26].

Limited multi-tasking

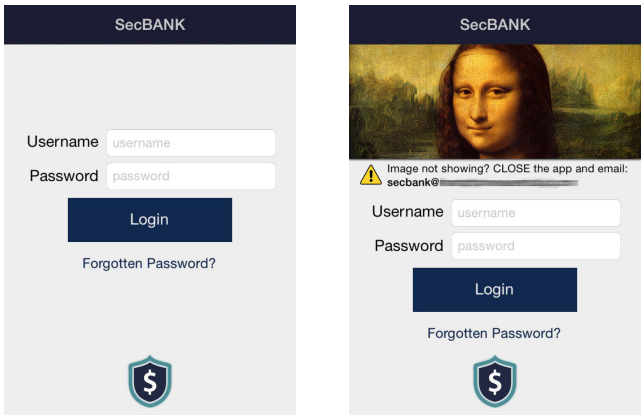
Another approach to counter background or floating attacks is to limit multi-tasking on the device. The legitimate application can trigger a restricted mode of operation where no third-party applications can activate to the foreground. Multi-tasking can be re-enabled once the user explicitly terminates the application. Activation to the foreground can always be allowed for system services, to receive phone calls or SMS messages. This approach does not rely on the user's alertness but it requires changes to the OS and hinders the user experience. For example, a user cannot receive social network notifications while he is interacting with an application that disables multi-tasking.

Application name

The mobile OS can show a status bar with the name of the application in the foreground [4, 35]. Phishing detection with this approach is effective only if the user is alert and the phishing application has a name and icon that are noticeably different from the ones of the legitimate application. This technique cannot address name similarity attacks. Furthermore, the status bar reduces the screen real estate for applications that run in full-screen mode. An approach where the status bar appears only when the user interacts with the application is only practical for applications with low interaction, such as video players (Android *Lean Back* mode). For applications that require constant interaction, such as games (Android *Immersive* mode), forcing a visible status bar would hinder the user experience.

Personalized indicator

When the application is installed, the user chooses an image from his photo gallery. When the application asks the user for his credentials, it displays the image chosen by the user at installation time. An alert user can detect a phishing attack if the application asking for his creden-



(a) (b)

Figure 1: (a) SecBank application for the baseline group. The application did not use personalized indicators. (b) SecBank application for the three experimental groups. The application displayed the personalized indicator chosen by the user on the login screen (i.e., the Mona Lisa).

tials does not show the correct image. The mobile OS prevents other applications from reading the indicator of a particular application (through application-specific storage). This countermeasure can also mitigate floating attacks. In particular, the legitimate application can check if it is running in the foreground and remove the image when it detects that the application has lost focus (e.g., overriding the `onWindowFocusChanged()` method). Personal indicators can be easily deployed as they do not require changes to the OS or to the marketplace. However, they demand extra user effort at install time (because the user must choose the indicator) and during application usage (because the user must check that the application displays the correct indicator).

Summary

Our analysis is summarized in Table 1. All the countermeasures we discuss incur trade-offs in effectiveness, usability, and deployment. Personalized indicators can address the many attack vectors and are easy to deploy as they do not require changes on the device or at the marketplace. However, personalized indicators rely on the user to detect phishing attacks.

Since smartphone applications are an increasingly important access method for many security-critical services such as e-banking; the user interface of mobile applications is significantly different from those of standard websites; and personalized indicators have not been evaluated in the context of smartphone applications, we decided to assess their effectiveness as a detection mechanism for mobile application phishing attacks.

USER STUDY

The goal of our user study was to evaluate the effectiveness of personalized indicators as a phishing-detection mechanism for mobile applications. We focused on a mobile banking scenario and implemented an application that allowed users to carry out e-banking tasks for a fictional bank called SecBank. As no bank currently uses indicators when the user performs a login operation, an evaluation in the context of a real deployment was not possible. The application had two components: a benign component that included the logic to log in and to carry out the banking tasks, and a malicious component that was in charge of carrying out the phishing attack. We used only one application to minimize the burden on the participants enrolling in our user study. That is, participants were asked to install only one application, rather than the banking application and another innocent-looking application that would perform the phishing attack.

In order to avoid participants focusing on the security aspects of the study, we advertised it as a user study to assess the usability of a mobile application. We asked participants to install the SecBank application on their phones and provided them with login credentials (username and password) to access their accounts at SecBank. We assigned each participant to either a baseline group that used a SecBank application without personalized indicators (Figure 1a), or one of three experimental groups that used it with personalized indicators (Figure 1b). The experimental groups differed by the type of phishing attack. The user study lasted one week. During the first three days, we asked participants to carry out one e-banking task per day, in order to familiarize participants with the application. On the seventh day, we asked participants to perform a fourth e-banking task and, at this time, the malicious component of the application performed a phishing attack. We recorded whether participants entered their credentials while under attack.

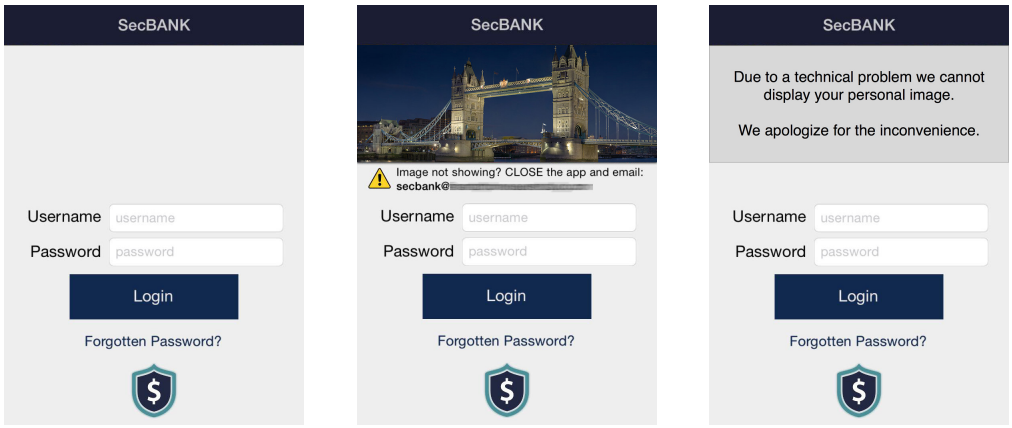
Ethical guidelines

We informed the participants that the application would record their input and have access to the photo gallery on their phones. We further explained that the application would send no personal information to our servers. We collected the participants' email addresses to send them instructions on how to complete the e-banking tasks. The email addresses were deleted once the study was finished. At the end of the study, we briefed participants about the true purpose and the methodology of the study. We notified the ethical board of our institution which reviewed and approved our protocol before we started the user study.

Procedure

Recruitment and group assignment

We recruited participants through an email sent to all people with an account at our institute (students, faculty and university staff). The study was advertised as a user study to "test the usability of a mobile banking application" without details of the real purpose of our design. We offered a compensation of $20 to all participants who completed the pre-test questionnaire.



(a) (b) (c)

Figure 2: (a) Missing-Image attack: The application does not show the indicator chosen by the user. (b) Random-Image attack: The application shows a random image from the local photo gallery (e.g., the Tower Bridge of London). (c) Maintenance attack: The application shows a message explaining that the indicator cannot be displayed due to technical reasons.

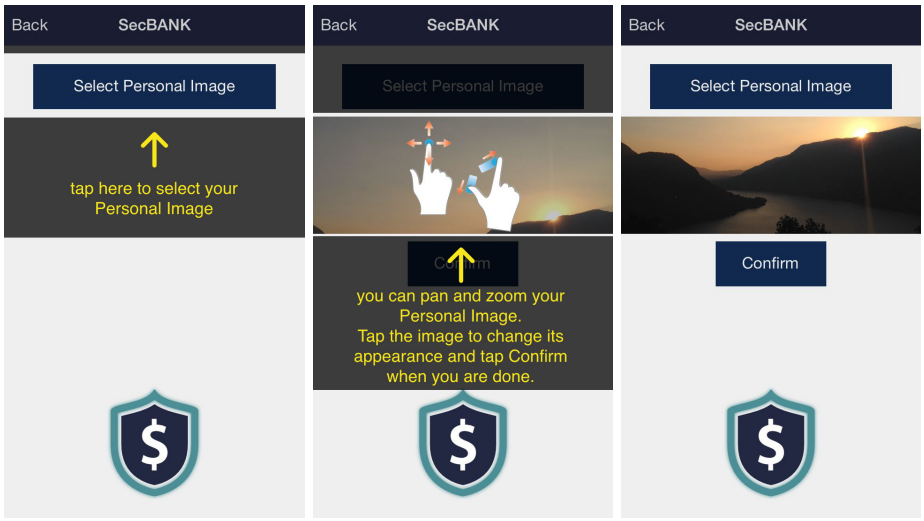
We received 465 emails from potential participants to whom we replied with a link to an online pre-test questionnaire designed to collect email addresses and demographic information. 301 participants filled in the pre-test questionnaire. We assigned them to the following four groups (one baseline group and three experimental groups) in a round-robin fashion:

- **Baseline Group (A):** The application used by this group did not use personalized indicators. On the last day of the user study, the malicious component of the application showed an exact clone of the SecBank login screen. The baseline group allows us to evaluate how many users, in a specific task, enter their credentials when shown a login screen that is identical to the legitimate one. We use the baseline login rate as a reference for comparing observed login rates in the experimental groups.
- **Missing-Image Group (B):** The application used by this group supported personalized indicators. On the last day of the user study, the malicious component of the application performed a phishing attack and showed the SecBank login screen without the indicator (Figure 2a).
- **Random-Image Group (C):** The application used by this group supported personalized indicators. On the last day of the user study, the malicious component of the application performed a phishing attack and showed the SecBank login screen with a photo randomly chosen from the local photo gallery. The photo displayed was different from the one chosen by the user as the personalized indicator (Figure 2b).
- **Maintenance Group (D):** The application used by this group supported personalized indicators. On the last day of the user study, the malicious component of the application performed a phishing attack and showed the SecBank login screen with an "under maintenance" notification in place of the indicator chosen by the user (Figure 2c).

We sent an email to all participants who completed the pre-test questionnaire with a link to a webpage from which they could install the SecBank application [27]. Participants in the Baseline Group (A) were directed to a webpage where we only explained how to install the application. Participants in experimental groups B, C, and D were directed to a webpage that also explained the concept of personalized indicators. The webpage advised that participants should not enter their login credentials if the application was not showing the correct indicator. The instructions were similar to the ones used in banking websites that deploy indicators [3,37]. For completenes, we report the full text shown to participants in the experimental groups.

"As a major banking institution, SecBank is committed to prevent fraudulent smartphone applications from stealing your password. The SecBank mobile application uses a novel login mechanism based on personal images. The first time you login, you will be asked to pick a personal image from the photos stored in your phone. From that moment, the SecBank application will display your personal image every time it asks for your username and password. The presence of the correct personal image guarantees that you are not using a fraudulent look-alike application. You should enter your username and password only when you see your personal image. (Your personal image remains on your phone and is not sent to our servers.)"

Figure 3 shows the screenshots of the SecBank application that were seen by participants in the experimental



(a) Step 1 (b) Step 2 (c) Step 3

Figure 3: Steps required to setup the Application Indicator in the SecBank application.

groups. The overlays (black boxes with yellow text) disappeared as soon as users interacted with the application. These screens were shown only *once* during the setup phase, at the beginning of our user study.

276 participants visited the webpages and installed the SecBank application on their devices. After installation, the SecBank application for groups B, C, and D showed explanatory overlays to guide participants in choosing a personalized indicator from their photo gallery.

Tasks

The study lasted one week. Participants were asked to perform four e-banking tasks on days 1, 2, 3, and 7. We sent instructions via email and asked participants to complete the task within 24 hours [27]. The tasks were the following: **Task 1** (Day 1): "Transfer $200 to Anna Smith"; **Task 2** (Day 2): "Download the bank statement from the Account Overview tab"; **Task 3** (Day 3): "Activate the credit card from the Cards tab"; **Task 4** (Day 7): "Transfer $100 to George White."

The goal of tasks 1–3 was to help participants to become familiar with the SecBank application. We sent the instructions to perform the last task four days after (including a weekend) the completion of task 3. During this last task, the malicious component of the application performed a phishing attack on all participants. Participants in the Baseline Group (A) saw a login screen that matched that of their SecBank application. Participants in the Missing-Image Group (B) saw a login screen similar to the one of SecBank, but without any personalized indicator (Figure 2a). Participants in the Random-Image Group (C) saw a login screen similar to SecBank, but with a random image from their photo gallery (e.g., the Tower Bridge as shown in Figure 2b). Finally, participants in the Maintenance Group (D) saw a message explaining that for technical problems the indicator could not be displayed (Figure 2c). In order to understand if participants fell for the phishing attack, during the last task, we recorded which users entered their credentials and which, instead, closed the application without entering their credentials.

Gender	
Male	150 (68%)
Female	71 (32%)
Age	
Up to 20	43 (20%)
21 – 30	164 (74%)
31 – 40	9 (4%)
41 – 50	3 (1%)
51 – 60	0 (0%)
Over 60	2 (1%)
Use smartphone to read emails	
Yes	214 (97%)
No	7 (3%)
Use smartphone for social networks	
Yes	218 (99%)
No	3 (1%)
Use smartphone for e-banking	
Yes	97 (44%)
No	124 (56%)

Table 2: Demographic information of the 221 participants that completed all tasks.

Results

Out of 276 participants that installed the application, 221 completed all tasks. We provide their demographics and other information collected during the pre-test questionnaire in Table 2. The majority of the participants were male (68%) and thirty years old or younger (94%). Most participants used their smartphone to read emails

	Attack not successful	Attack successful
Baseline Group (A)	0 (0%)	56 (100%)
Missing-Image Group (B)	30 (55%)	25 (45%)
Random-Image Group (C)	23 (41%)	33 (59%)
Maintenance Group (D)	29 (54%)	25 (46%)
Experimental groups combined	82 (50%)	83 (50%)

Table 3: Success rate of the phishing attack.

	Attack not successful	Attack successful
Gender		
Male	59 (52%)	54 (48%)
Female	23 (44%)	28 (56%)
Age		
Up to 20	15 (43%)	20 (57%)
21 – 30	57 (48%)	61 (52%)
31 – 40	6 (86%)	1 (14%)
41 – 50	2 (67%)	1 (33%)
51 – 60	0 (0%)	0 (0%)
Over 60	2 (100%)	0 (0%)
Use smartphone for e-banking		
Yes	41 (54%)	35 (46%)
No	41 (46%)	48 (54%)
Smartphone display size (diagonal)		
up to 4in	28 (58%)	20 (42%)
from 4in to 4.5in	44 (45%)	54 (55%)
from 4.6in to 5in	10 (53%)	9 (47%)

Table 4: Success rate of the phishing attack in relation to gender, age, familiarity with mobile banking, and smartphone display size.

(97%) and to access social networks (99%). Slightly less than half of the participants (44%) used their smartphones for mobile banking.

The 221 participants that completed all tasks were distributed as follows: 56 in the Baseline Group (A), 55 in the Missing-Image Group (B), 56 in the Random-Image Group (C), and 54 in the Maintenance Group (D).[1]

Indicator effectiveness

Table 3 shows the success rates for the phishing attack during Task 4. All of the 56 participants in the Baseline Group (A) entered their login credentials. 83 out of 165 (50%) attacks in the experimental groups B, C, and D were successful.

To analyze the statistical significance of these results we used the following null hypothesis: "there will be no difference in the attack success rate between users that use personalized indicators and users that do not use personalized indicators". A Chi-square test showed that the difference was statistically significant ($\chi^2(1, N = 221) = 44.25, p < 0.0001$) and thus the null hypothesis can be rejected. We conclude that, in our user study, the deployment of security indicators decreased the attack success rate and improved phishing detection.

Difference between attacks

A closer look at the performance of participants in groups B, C, and D reveals that: 30 out of 55 participants in the Missing-Image Group (B), 23 out of 56 participants in the Random-Image Group (C), and 29 out of 54 participants in the Maintenance Group (D) did not log in.

To analyze the success rates of the different attack types we used the following null hypothesis: "the three attack types we tested are equally successful". A Chi-squared test showed no statistically significant difference in the attack success rates, and thus we fail to reject the null hypothesis ($\chi^2(2, N = 165) = 2.53, p = 0.282$).

Other factors

We performed post hoc analysis of our dataset to understand if there were any relationship between the attack success rate and gender ($\chi^2(1, N = 221) = 0.99, p = 0.319$), age group ($\chi^2(4, N = 221) = 8.36, p = 0.079$), smartphone display size ($\chi^2(2, N = 221) = 5.40, p = 0.369$) or familiarity with mobile banking ($\chi^2(1, N = 221) = 1.98, p = 0.160$). We did not find any statistical significance for any of the factors we considered. Table 4 provides the results break-down.

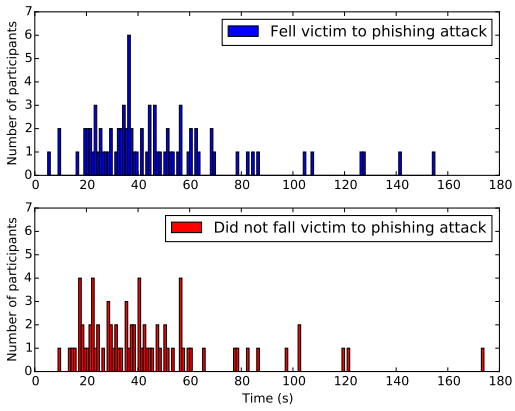
Finally, we report the mean time spent by participants setting up the personalized indicator or logging in. The mean time spent setting up the indicator for participants that did not fall victim to the attack was 43s (±28s); the mean time for participants that fell for the attack was 46s (±28s). The mean time spent on the login screen for participants that did not fall victim to the attack was 18s (±14s); the mean time for participants that fell for the attack was 14s (±10s). The distribution of the times spent while setting up the indicator and while logging in are shown in Figure 4a and Figure 4b, respectively.

Post-test questionnaire

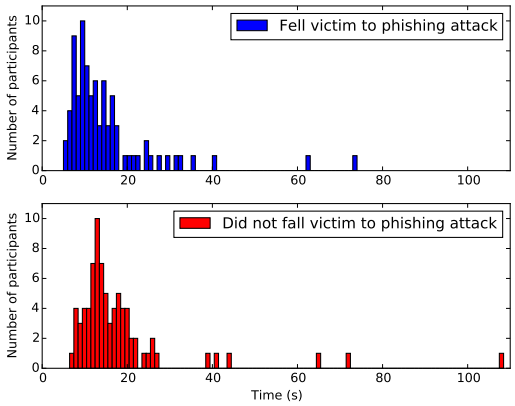
At the end of the user study we asked participants to complete a short post-test questionnaire. In particular we asked participants in the experimental groups if they were familiar with security indicators prior to our user study and only 19% replied that they were.

We also asked them if they had noticed anything unusual when logging in to complete Task 4 (the simulated phishing attack). 23% of the participants did not notice anything unusual, while 23% did not remember. 54% of the participants noticed something was wrong with the SecBank application while they were logging in. To those

[1]We note that, by chance, the participants that dropped out of the study were almost evenly distributed among the four groups.



(a) Personal indicator setup



(b) Login screen

Figure 4: Distribution of the time spent by participants setting up the personalized indicator (a) and logging into the SecBank application (b).

participants that noticed something wrong with the application, we also asked if they logged in and why (answer to this question was not mandatory.) 36% of them reported that they logged in and the reasons they provided mainly fell in two categories. Some users reported that they logged in because they were role-playing and did not have anything to lose. We list some of their answers below:

"Because it is a test and I had nothing to lose or hide."

"This should be a test and I thought that it was safe."

"I knew this was a test product so there could not possibly be malware for it already. I just thought maybe you guys had some difficulties going on."

Some users from the Maintenance Group (group D) reported that they logged in because they thought there was a temporary bug in the application. This behaviour suggests that users expect bugs in IT systems and, therefore, they are susceptible to attacks that leverage the unreliable view that people have of computer products. We list some of participants' answers below:

"I thought it was a problem of the app that the image *was* there but just did not load. As it happens sometimes in Safari or other browsers."

"I thought it was a temporary bug."

"I thought that it was a system error."

DISCUSSION

In our study, the deployment of security indicators prevented half of the attacks. Our user study shows a significant improvement in the attack detection rate (50%), compared to previous studies in the context of website phishing attacks (4% in [33] and 27% in [23]). The purpose of our study was not to reproduce these previous studies but rather to evaluate security indicators in the context of smartphone applications as realistically as possible. Below we discuss how our results should be interpreted in comparison to previous related studies and outline directions for further studies that are needed to gain better confidence on the effectiveness of personalized indicators.

Role-playing

When designing our user study we kept the experiment as close to a real world deployment as possible. We asked participants to install the application on their phones and avoided using web platforms for human intelligence tasks like Amazon Mechanical Turk (used in, for example, [23]). However, we could not leverage a real bank deployment and its user base (as in [33]) because, to the best of our knowledge, no bank is currently using personalized indicators in its mobile banking application.

Previous work has shown that role-playing negatively impacts the effect of security mechanisms [33]. The responses to the post-test questionnaire give reasons to believe that, due to role-playing, some participants may have logged in despite detecting the attack. It is likely that role-playing increased the attack success rates in our study. A user study run in cooperation with a banking institution willing to deploy personalized indicators would yield more accurate results.

Duration and security priming

In our study, the phishing attack happened seven days after the study participants had been primed about security and our study did not evaluate participants' behavior at a later point in time. This study setup is similar to Lee et al. [23] work, where 5 days passed between participants priming and the attack. In contrast, Schechter et al. [33] recruited customers of a real bank that had been primed at the time they had opened their bank account (possibly long before they took part in the user study and the attack was tested). It is likely that compared to a real-world deployment, the recent security priming of our user study decreased the attack success rates. Long-term studies in the context of mobile applications (potentially through a real-bank deployment) are needed to evaluate the effect of time between the security priming and the attack.

Population sample

Participants were recruited within our institution across students, faculty and staff. Most participants were male (68%) and below 30 years old (94%). While our institution attracts people from around the world, the large majority of the participants were Swiss nationals. As our institution has 16 departments we reached many participants that don't have a computer security background. Also, many mobile banking users are relatively young [5], thus our sample overlaps with the expected user population. Further studies are nonetheless required to assess whether our results generalize to different populations (e.g., with different age intervals, nationalities, etc.).

We did not ask participants whether they knew other participants and whether they had discussed the study. While participants may influence each other's behavior, we could not identify any particular relationship or cliques among participants.

Application deployment

In our study, we distributed the victim application (e-banking app) and the phishing component in a single application, rather than using one victim application and a second application to launch the attack. Since the phishing attack was not launched from a separate application, our study did not evaluate whether participants could detect the attack by UI lag when the phishing application gains control of the device screen.

The motivation behind this study design choice was twofold. First, we minimized the participant burden during enrollment. Participants were asked to install only the SecBank application, rather than the banking application and a second innocent-looking application that would launch the attack. If participants had had to install a second application (e.g., a weather forecast application) they may have become suspicious about its purpose. Second, previous work has shown that users tend to disregard slight animation effects when the phishing application gains control of the device screen [4]. Due to this design choice, if a study participant decided to examine the list of running background apps before entering his login credentials, our attack component would not have been visible on this list, and thus such defensive measures are not applicable to our study.

Recruitment and task perception

A common challenge in designing security-related user studies is to avoid drawing participants' attention to the security aspects under evaluation. If participants are focused on security, and hence more attentive to possible threats, the study results would say little about real-world users to whom security is typically not the primary goal [13, 33]. As our goal was to assess the effectiveness of a security mechanism that has not yet been deployed in the context of smartphone applications, we could not avoid minimal security priming of the participants.

We advertised our study as one on "the usability of a mobile banking application". Similarly, the emails sent to complete the tasks were solely focused on task completion [27]. We cannot verify if some participants discovered the true goal of our study before we revealed it. However, the comments that participants entered in the post-test questionnaire suggest that many participants focused on the usability of the application. We report some comments we received:

"The tasks were easy to perform, but it remained unclear for me what you were exactly testing."

"App easy to navigate and user-friendly."

"The user interface was not so intuitive due to the lack of spaces between buttons and the equality of all interface options/buttons."

Attack implementation

In the phishing attacks where the UI showed no indicator (group B) or where it showed a maintenance message (group D), we removed the text that asked users to email the bank in case of a missing indicator. We kept that text in the attack that showed a random image (group C). The UI elements shown by the phishing application might have influenced the reaction of the participants and their willingness to enter their credentials. We did not test how changes to the text or to other UI elements affect phishing detection. A potential direction for future studies is to understand how users react to small changes to the UI of an application.

Indicator placement and size

The SecBank application showed the personalized indicator right above the username and password fields, taking up roughly one third of the screen. The size and the placement of the personalized indicator within the UI may have an impact on the attack detection rate. In the context of websites designed for PC platforms, Lee et al. [23] show that the size of the indicator does not change the effectiveness of personalized indicators as a phishing-detection mechanism. An interesting direction for future work would be to look at alternative types of indicators (e.g., interactive ones) and compare them to the ones used in this work.

DEPLOYMENT ASPECTS

Application and infrastructure changes

From the point of view of a service provider, personalized indicators can be easily deployed because they require no changes to the marketplace or to the mobile OS. Introducing personalized indicators only requires a software update of the client application (application updates are frequent throughout an application lifecycle) and no changes to the server-side infrastructure of the application provider (i.e., the bank). The mobile application may guide the user through the indicator setup. Other solutions, as those presented earlier on, require either changes to the mobile OS or to the marketplace infrastructure. A service provider (e.g., a bank) can therefore adopt this security mechanism independently of other service providers or of the mobile platform provider.

Indicator choice and reuse

Personalized indicators may be used for phishing detection by security-critical applications. If indicators are adopted by multiple applications, users might tend to reuse the same indicator across different applications. This behaviour may provide an attack vector where the attacker develops an application that requires personalized indicators, and hopes that the victim user chooses the same indicator that he had chosen for his banking application. The problem of reusing personalized indicators across applications is comparable to the problem of reusing passwords across online services. We note that the deployment of personalized indicators would most likely be limited to few security-critical services, while users often have to manage passwords for a large number of services and websites.

Similar to password reuse scenarios, users might choose different personalized indicators for "categories" of applications. That is, a particular picture for security critical applications (e.g., banking, email) and another picture for less critical applications (e.g., social networks). Furthermore, when users are asked to pick a personalized indicator, they might choose among the pictures that are at the top of the list (e.g., the ones that were most recently added to the photo gallery). Therefore, the probability that a picture is selected as the personalized indicator may not be uniform across all pictures in the photo gallery.

Since in our study we did not collect information on the indicators chosen by the participants, further studies are required to explore users' behavior and patterns in choosing personalized indicators.

RELATED WORK

Mobile application phishing attacks have been described in recent research [4, 7, 15, 40] and several attacks have been reported in the wild [11, 14, 34]. Proposed countermeasures are primarily attack specific, i.e., they identify an attack vector and try to restrict or monitor access to the device functionality that enables the exploit [7, 21, 40].

A systematic evaluation of application phishing attacks was recently provided in [4]. The authors use static analysis to detect applications using APIs that enable certain classes of application phishing attacks. They also introduce an on-device solution that allows users to identify applications with which they are interacting. In the proposed solution, the OS displays a status bar that shows the application and developer names together with an image chosen by the user. The image, therefore, is used by the user to distinguish the authentic status bar managed by the OS from a fake status bar that a phishing application can show if it gains control of the entire screen. Compared to personalized indicators, the proposed solution incurs more deployment costs, since it requires changes to the OS and the marketplace. The authors of [4] also use Amazon Mechanical Turk to run a user study with 304 participants, and assess the effectiveness of phishing attacks in mobile platforms. The user study corroborates our findings on personalized indicators, although the authors placed the image in the navigation bar rather than in the application itself. Furthermore, the user study in [4] was a one-off test that did not last for a week and, compared to ours, let participants interact with an emulated Android device through a web-browser rather then letting participants use their phones in their own typical setting.

Several anti-phishing mechanisms have been proposed (and also deployed) for the web. Countermeasures include automated comparison of website URLs [28], visual comparison of website contents [6, 41], use of a separate and trusted authentication device [30], personalized indicators [9, 23, 33], multi-stage authentication [19], and attention key sequences to trigger security checks on websites [39]. Despite the many proposed countermeasures, web phishing remains an open problem [10, 20]. While some of these mechanisms are specific to the web environment, others could be adapted also for mobile application phishing detection. Website phishing in the context of mobile web browsers has been studied in [29, 32].

Previous research on the effectiveness of security indicators has mostly focused on phishing and SSL warnings on the web. Studies in this context have shown that users tend to ignore security indicators such as personalized images [23, 33] or toolbars [22, 38]. Browser security warnings (e.g., for an invalid server certificate) have been shown to be effective on recent browser versions [1], while previous studies on older browser warning implementations found the security warnings ineffective [10, 12, 36].

CONCLUSION

Phishing attacks are an emerging threat for mobile application platforms and the first successful attacks have already caused significant financial losses. Personalized indicators are a well-known countermeasure to address the problem of phishing, but previous studies in the context of websites have shown that indicators fail to prevent the majority of attacks. In this paper we report our findings from the first user study on smartphones that evaluates the effectiveness of personalized security indicators for mobile applications. Our preliminary results show that in the new context of smartphones applications, personalized indicators could help users detecting application phishing attacks.

We conclude that personalized indicator can be an effective mean to thward application phishing attacks and further studies are needed to fully understand their benefits in new deployment models such as in mobile applications.

REFERENCES

1. Devdatta Akhawe and Adrienne Porter Felt. 2013. Alice in Warningland: A Large-scale Field Study of Browser Security Warning Effectiveness. In

USENIX Conference on Security (USENIX'13). 257–272.

2. Tanzirul Azim and Iulian Neamtiu. 2013. Targeted and Depth-first Exploration for Systematic Testing of Android Apps. In *International Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA'13)*. 641–660.
3. Bank of America. 2006. SiteKey Authentication. (2006). https://www.bankofamerica.com/privacy/online-mobile-banking-privacy/sitekey.go (last access 2015).
4. Antonio Bianchi, Jacopo Corbetta, Luca Invernizzi, Yanick Fratantonio, Christopher Kruegel, and Giovanni Vigna. 2015. What the App is That? Deception and Countermeasures in the Android User Interface. In *IEEE Symposium on Security and Privacy (S&P'15)*.
5. Board of Governors of the Federal Reserve System. 2015. Consumers and Mobile Financial Services 2015. (2015). http://www.federalreserve.gov/econresdata/consumers-and-mobile-financial-services-report-201503.pdf (last access 2016).
6. Teh-Chung Chen, Scott Dick, and James Miller. 2010. Detecting Visually Similar Web Pages: Application to Phishing Detection. *ACM Trans. Internet Technol.* 10, 2 (2010), 1–38.
7. Erika Chin, Adrienne Porter Felt, Kate Greenwood, and David Wagner. 2011. Analyzing Inter-application Communication in Android. In *International Conference on Mobile Systems, Applications, and Services (MobiSys'11)*. 239–252.
8. Thomas Davies and Ashweeni Beeharee. 2012. The Case of the Missed Icon: Change Blindness on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*. 1451–1460.
9. Rachna Dhamija and J. D. Tygar. 2005. The Battle Against Phishing: Dynamic Security Skins. In *Symposium on Usable Privacy and Security (SOUPS'05)*. 77–88.
10. Rachna Dhamija, J. D. Tygar, and Marti Hearst. 2006. Why Phishing Works. In *SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*. 581–590.
11. Digital Trends. 2013. Do not use iMessage chat for Android, it's not safe. (2013). www.digitaltrends.com/mobile/imessage-chat-android-security-flaw/ (last access 2016).
12. Serge Egelman, Lorrie Faith Cranor, and Jason Hong. 2008. You'Ve Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. 1065–1074.
13. Serge Egelman, Jennifer King, Robert C. Miller, Nick Ragouzis, and Erika Shehan. 2007. Security User Studies: Methodologies and Best Practices. In *Extended Abstracts on Human Factors in Computing Systems (CHI EA'07)*. ACM, 2833–2836.
14. F-secure. 2010. Warning On Possible Android Mobile Trojans. (2010). www.f-secure.com/weblog/archives/00001852.html (last access 2016).
15. Adrienne Porter Felt and David Wagner. 2011. Phishing on Mobile Devices. In *Web 2.0 Security and Privacy Workshop (W2SP'11)*. 1–10.
16. Forbes. 2015. Alleged "Nazi" Android FBI Ransomware Mastermind Arrested In Russia. (2015). http://www.forbes.com/sites/thomasbrewster/2015/04/13/alleged-nazi-android-fbi-ransomware-mastermind-arrested-in-russia/ (last access 2016).
17. Google Inc. 2012. Android and Security. (2012). http://googlemobile.blogspot.ch/2012/02/android-and-security.html (last access 2016).
18. Google Inc. 2016. Protect against harmful apps. (2016). https://support.google.com/accounts/answer/2812853 (last access 2016).
19. Amir Herzberg and Ronen Margulies. 2012. My Authentication Album: Adaptive Images-Based Login Mechanism. In *Information Security and Privacy Research*. 315–326.
20. Jason Hong. 2012. The state of phishing attacks. *Commun. ACM* 55, 1 (2012), 74–81.
21. Jie Hou and Qi Yang. 2012. Defense Against Mobile Phishing Attack. (2012). www-personal.umich.edu/~yangqi/pivot/mobile_phishing_defense.pdf (last access 2016).
22. Collin Jackson, Daniel R Simon, Desney S Tan, and Adam Barth. 2007. An evaluation of extended validation and picture-in-picture phishing attacks. In *Financial Cryptography and Data Security*. 281–293.
23. Joel Lee, Lujo Bauer, and Michelle L. Mazurek. 2014. Studying the effectiveness of security images in Internet banking. *IEEE Internet Computing* 13, 1 (2014).
24. Chia-Chi Lin, Hongyang Li, Xiaoyong Zhou, and XiaoFeng Wang. 2014. Screenmilker: How to Milk Your Android Screen for Secrets. In *Network and Distributed System Security Symposium (NDSS'14)*.
25. MacRumors. 2014. 'Masque Attack' Vulnerability Allows Malicious Third-Party iOS Apps to Masquerade as Legitimate Apps. (2014). http://www.macrumors.com/2014/11/10/masque-attack-ios-vulnerability/ (last access 2016).

26. Luka Malisa, Kari Kostiainen, and Srdjan Capkun. 2015. Detecting Mobile Application Spoofing Attacks by Leveraging User Visual Similarity Perception. (2015). Cryptology ePrint Archive: Report 2015/709.

27. Claudio Marforio, Ramya Jayaram Masti, Claudio Soriente, Kari Kostiainen, and Srdjan Capkun. 2016. Supplementary Material - Evaluation of Personalized Security Indicators as an Anti-Phishing Mechanism for Smartphone Applications. (2016). http://www.smartphoneuserstudy.com/supplementary_chi16.pdf (last access 2016).

28. Max-Emanuel Maurer and Lukas Höfer. 2012. Sophisticated Phishers Make More Spelling Mistakes: Using URL Similarity against Phishing. In *International Conference on Cyberspace Safety and Security (CSS'12)*. 414–426.

29. Yuan Niu, Francis Hsu, and Hao Chen. 2008. iPhish: Phishing Vulnerabilities on Consumer Electronics. In *Conference on Usability, Psychology, and Security (UPSEC'08)*. 1–8.

30. Bryan Parno, Cynthia Kuo, and Adrian Perrig. 2006. Phoolproof Phishing Prevention. In *International Conference on Financial Cryptography and Data Security (FC'06)*. 1–19.

31. Ronald A Rensink. 2002. Change detection. *Annual review of psychology* 53, 1 (2002), 245–277.

32. Gustav Rydstedt, Baptiste Gourdin, Elie Bursztein, and Dan Boneh. 2010. Framing Attacks on Smart Phones and Dumb Routers: Tap-jacking and Geo-localization Attacks. In *USENIX Workshop on Offensive Technologies (WOOT'10)*. 1–8.

33. Stuart E. Schechter, Rachna Dhamija, Andy Ozment, and Ian Fischer. 2007. The Emperor's New Security Indicators. In *IEEE Symposium on Security and Privacy (S&P'07)*. 51–65.

34. Secure List. 2013. The Android Trojan Svpeng now capable of mobile phishing. (2013). www.securelist.com/en/blog/8138/The_Android_Trojan_Svpeng_now_capable_of_mobile_phishing (last access 2016).

35. Marcel Selhorst, Christian Stüble, Florian Feldmann, and Utz Gnaida. 2010. Towards a trusted mobile desktop. In *International Conference on Trust and Trustworthy Computing (TRUST'10)*. 78–94.

36. Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor. 2009. Crying Wolf: An Empirical Study of SSL Warning Effectiveness.. In *USENIX Security Symposium*. 399–416.

37. The Vanguard Group. 2006. Vanguard Enhanced Logon. (2006). http://www.vanguard.com/us/content/Home/RegEnhancedLogOnContent.jsp (last access 2015).

38. Min Wu, Robert C. Miller, and Simson L. Garfinkel. 2006a. Do Security Toolbars Actually Prevent Phishing Attacks?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. 601–610.

39. Min Wu, Robert C. Miller, and Greg Little. 2006b. Web Wallet: Preventing Phishing Attacks by Revealing User Intentions. In *Symposium on Usable Privacy and Security (SOUPS'06)*. 102–113.

40. Zhi Xu and Sencun Zhu. 201. Abusing Notification Services on Smartphones for Phishing and Spamming. In *USENIX Workshop on Offensive Technologies (WOOT'12)*. 1–11.

41. Yue Zhang, Jason I. Hong, and Lorrie F. Cranor. 2007. Cantina: A Content-based Approach to Detecting Phishing Web Sites. In *International Conference on World Wide Web (WWW'07)*. 639–648.

42. Yajin Zhou, Zhi Wang, Wu Zhou, and Xuxian Jiang. 2012. Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets. In *Network and Distributed System Security Symposium (NDSS'12)*.