An empirical approach to phishing countermeasures through smart glasses and validation agents

Jema David Ndibwile, Edith Talina Luhanga, Doudou Fall, Daisuke Miyamoto, Gregory Blanc, Youki Kadobayashi

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

An Empirical Approach to Phishing Countermeasures through Smart Glasses and Validation Agents

JEMA DAVID NDIBWILE[1], EDITH TALINA LUHANGA[2], DOUDOU FALL[1], DAISUKE MIYAMOTO[4], GREGORY BLANC[3], AND YOUKI KADOBAYASHI.[1], (IEEE, Member)

[1]Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara, Japan, 630-0192
[2]Nelson Mandela African Institution of Science and Technology, P.O.Box 447, Arusha, Tanzania
[3]Samovar, CNRS, Télécom SudParis, Institut Polytechnique de Paris, 9 rue Charles Fourier, 91011 Evry Cedex, France
[4]Tokyo University, 7 Chome-3-1 Hongo, Bunkyo City, Tokyo, Japan 113-8654

Corresponding author: Jema David Ndibwile (e-mail: jema.ndibwile.je8@is.naist.jp or jemablue86@gmail.com).

"This work was supported by JSPS KAKENHI Grant Number JP17K00180."

ABSTRACT Phishing attacks have been persistent for more than two decades despite mitigation efforts from academia and industry. We believe that users fall victim to attacks not only because of lack of knowledge and awareness, but also because they are not attentive enough to security indicators and visual abnormalities on the webpages they visit. This is also probably why smart device users, who have more limited screen size and device capabilities compared to desktop users, are three times more likely to fall victim to phishing attacks. To assert our claim, we first investigated general phishing awareness among different groups of smartphone users. We then used smart eyeglasses (electro-oculographic) to experimentally measure the mental effort and vigilance exhibited by users while surfing a website and while playing an Android phishing game that we developed. The results showed that knowledge and awareness about phishing do not seem to have a significant impact on security behaviours, as knowledgeable participants exhibited insecure behaviours such as opening email attachments from unfamiliar senders. However, attentiveness was important as even participants with low cybersecurity knowledge could effectively identify attacks if they were reasonably attentive. Based on these results, we asserted that users are more likely to continue falling victim to phishing attacks due to insecure behaviours, unless tools to lessen the identification burden are provided. We thus recommended implementing a lightweight algorithm into a custom Android browser for detecting phishing sites deceptively without a user interaction. We used fake login credentials as validation agents and monitor the destination server HTTP responses to determine the authenticity of a webpage. We also presented initial evaluation results of this algorithm.

INDEX TERMS Social engineering, Phishing, Smart glasses, Mobile devices, Deceptive login, Android browser, Electro-oculographic, Cybersecurity psychology.

I. INTRODUCTION

THE increase in the use of smart devices such as smartphones and the increasing amount of important information they store make them a prime target by attackers interested in exploiting them [1]. The exploitation can either be through the device itself or a human. The most common and classic human exploitation is phishing, which mostly relies on the naivety of users. Phishing is an art of deception where a user is fraudulently convinced by an attacker to divulge sensitive information such as credit card or login credentials. The attacks are launched via electronic communications *e.g.*, emails and instant messages purporting to be from a trusted contact, and the victims are deceived to click on a malicious link. Clicking the link can result in installation of malicious software (malware), locking of the user's computing devices (ransomware) or redirection to a fake website.

The number of phishing attacks is increasing every year [2]. Research shows that despite the persistence of cyber-attacks that rely on the naivety of users *e.g.*, phishing, cybersecurity knowledge among users is globally still low [3]. Several mitigation strategies have been proposed [4] [5] [7] [8], but preventing attacks that target user-application inter-

actions is still a major challenge as the deception techniques used and the fake websites deployed are increasingly sophisticated. Falling victim to attacks when using mobile devices is also three times more likely than on desktops [6] due to the devices' limited screen size and capabilities. Limited computational power, for instance, limited the number of PC solutions that can be implemented on smart devices. A simple example is the omission of mobile phone browser plugins. Likewise, web features such as URL bar are not spacious enough to accommodate a lengthy URL address which is normally truncated and, sometimes, hidden. Regardless of these additional challenges, little has been done to address this. As smartphone penetration and internet usage via mobile devices continue to rise, it is important to understand users' cybersecurity behaviours and what factors are most responsible for cybersafety behaviours.

In this study, we aimed to investigate whether the role of users' cybersecurity knowledge level, attention and vigilance to security indicators displayed on smartphone web browsers have an impact on their abilities to identify phishing (fake) websites from legitimate ones. We first conducted a survey with 206 participants to determine their self-reported cybersecurity knowledge, attitudes and behaviours, and to objectively determine how accurate their self-reported knowledge levels were. We then sampled 40 participants (50% computer scientists (tech-savvy) and 50% non-computer scientists (non-tech-savvy) from different demographics) and assessed the mental effort they used when viewing a mixture of fake and legitimate websites via an Android game that we developed. Mental effort was estimated through smart eyewear (JINS MEME glasses) which can detect eye and body movement and horizontal and vertical electrooculogram (EOG) signals which are amplified by metal dry electrodes near the nasion and rhinion parts of the nose and eye [9]. In the game, we kept count of the number of correctly identified phishing websites. We hypothesized that while cybersecurity professionals would easily spot phishing sites or emails, average users would have varying degrees of success depending on their vigilance.

To summarize, our research objectives were:

1) To examine security behaviours of users pertaining to phishing.
2) To examine the impact of mental effort in identifying phishing attacks on websites and emails.
3) To use the results of objectives number 1 and 2 to improve our previous solution for phishing detection and intervention.

Our findings for the first study revealed that a majority of the participants are aware of security issues but it does not translate into security-conscious behaviours. For instance, we noticed that education level does not significantly influence the behaviours of users in phishing attacks. In our second study, we observed some positive correlations between the estimated mental effort of the users and phishing attack identification. The correlations among tech-savvy participants were more consistent and positive than non-tech-savvy participants. The tech-savvy participants who exhibited higher mental efforts had a higher phishing identification rate than those with lower or average efforts. However, despite the correlations among non-tech-savvy participants not being as strong as the tech-savvy participants, it was still positive and linear.

Based on these findings, we believe that users in general are more likely to continually fall victim to phishing attacks due to insecure behavior and negligence regardless of being educated or aware of the attack. From a usable security viewpoint, we believe that the burden to identify phishing attacks should not entirely be borne by the users. Thus, to help smart device users to browse safely online, we recommend an Android application prototype that can be incorporated inside a native Android browser. The prototype uses dummy login credentials to thwart phishing sites, as phishing sites almost always allow users to proceed to the next page even if the credentials are wrong [44]. Incorporating the prototype inside a native browser means the user does not need to install third-party security plugins or download another standalone app. It also removes the necessity of the user to manually check whether a website is a phishing one.

II. BACKGROUND AND RELATED WORK

This section highlights some works that analyze human cognitive ability through eye movements on performing certain tasks. It also describes some previous works that show the usefulness of validation agents such as dummy login credentials and how a user can avoid entering web-login credentials manually.

A. WEB LOGIN AUTOMATION

To avoid entering web-login credentials, a user of a smart device such as smartphone can authenticate herself to another device such as a laptop or another mobile device through a close-range Bluetooth communication. This strategy prevents phishing attacks as it uses a multi-factor authentication.

Han et al. introduced a Bluetooth-enabled smart device as a platform to store the login information of a user such as ID and password [14]. A smart device pre-stores information features of a login user interface. Then before the user enters the authentication information in another device, a plugin on her web browser communicates with the smart device through Bluetooth to verify the login credentials. After passing the login credential verification, the smart device automatically fills the login information to the login page on behalf of the user. However, for performance issues most smartphone browsers are not equipped with plugins.

Similarly, *Bridge* et al. introduced a method for automatically submitting login credentials, seamlessly, for a user of a web service [15]. The login information and credentials corresponding to the login form of a web service are stored and then used to authenticate the user for a session of the web service. A login token, generated by the web service, and its expiration date are tracked. The login credentials are then

automatically submitted, without user intervention, to the web service based on the login endpoint and the expiration date of the login token. The challenge with these solutions is the requirement of an extra Bluetooth-enabled device and browser plugins which are computationally expensive for a small device.

Another important approach to address phishing attacks is by testing the trustworthiness of the login webpage by injecting random login inputs into the webpage form fields. However, most of these works are browser-based plugins, and particularly designed to Mozilla Firefox browser. *Yue* et al. proposed Firefox browser-based solution to protect against phishing attacks with bogus bites [5]. A Firefox browser extension transparently inputs a relatively large number of bogus credentials into a suspicious phishing website, rather than attempts to prevent vulnerable users from browsing it. These bogus bites conceal victims' real credentials among bogus credentials and enable legitimate websites to identify stolen credentials in a timely manner. However, the installation must be done at both client and server side. Users need to install BogusBiter and a legitimate server needs to deploy the defensive line enabled by BogusBiter. Moreover, the other concern regarding a massive deployment of BogusBiter is that if the login page of a legitimate website is wrongly flagged as a phishing page, the load on the authentication web server will increase significantly due to a large number of bogus bites.

Shahriar et al. proposed and implemented a desktop-based testing tool named Phishtester [7]. Phishtester works by testing the trustworthiness of a number of suspicious websites through a provision of unknown random inputs to the login page. The tester checks the login page response against the pre-established known symptoms for a malicious site. This solution uses Finite State Machine (FSM) logic and only works based on a trigger that depends on certain conditions, signifying phishing attempt, to be true as pre-specified. Like any other rule-based system, if all the antecedent(s) of a rule are true, then the system is triggered. This might not be suitable to all problem domains, it is only suitable when a system behavior can be decomposed into separate states with well-defined conditions for state transitions [16]. *Wu* et al. proposed MobiFish [4] mobile application, which is implemented through Optical Character Recognition (OCR) for checking visual similarities between malicious and legitimate websites. However, visual similarity approaches have a tendency of missing well-presented phishing webpages [17].

Briefly, the discussed related approaches were mainly implemented in PC in which they require huge amount of computational power that might completely not work in small-sized devices or cause undesirable user experiences. Some features that might be necessary for security functionality for PCs are removed in mobile devices in order to accommodate their limited computational resources. For instance, browser plugins for mobile devices are often omitted or ignored due to their resource drainage. Browser developers point of view for not including plugins and extensions is the unmanageable consumption of resources (CPU/GPU and RAM). When these plugins are active on mobile devices they may lead to an uncontrollable browsing experience and power consumption. Furthermore, these works require intensive analysis of the features to identify malicious sites including the dependence on certain rules (rule based) to be true or certain amount of traffic for them to work.

B. MENTAL EFFORT ON PHISHING IDENTIFICATION

The idea of analyzing eye movements for correlating them with some cognitive tasks has recently been gaining pace, influenced by smart-eyewear technologies. "*Eye can tell*" by *Miyamoto* et al. [10] is one of the classic work of eye-movement analysis for estimating the attentiveness of users on phishing webpages, however it was carried out on a PC environment equipped with an eye-tracking device, and not using an eyewear. Thus, it is neither suitable for small devices nor capable of revealing a user response to changes in her mind and body. The experiment was solely limited to the use of a display device (computer) for verifying what kind of information a user checks on a website for phishing identification.

Similarly, a strong correlation between eye movements and activities such as reading has been observed in several research where the number of words a user reads is estimated [11]. However, the estimation is solely based on a powerful computer and a display device which highly limits its implementation in several environments. In one of the recent works, *Kunze* et al. implemented EOG-based technique with JINS MEME smart glasses to estimate the word counts read by a user [12]. EOG-based technique through devices like JINS MEME smart glasses has also been used in a range of unobtrusive activity tracking. *Ishimaru* et al. used an early prototype of JINS MEME smart glasses, and demonstrated how simple eye-movement visualization, body posture, and eye blink can be used to recognize and analyze certain human activity patterns such as talking, reading, and walking [13].

From this perspective, the cognitive effort can be estimated through the eye movements when a user is trying to identify a phishing website. The information obtained can be used by the solution developers to gain an insight on what proper security indicators to add or omit into the existing anti-phishing tools.

III. PHISHING AND PRIVACY KNOWLEDGE OF USERS

This section describes how we conducted our first study, from data collection to analysis and presents the findings.

A. STUDY METHOD

We conducted a survey with 206 people (62.35% male, 37.65% female) of varying education levels. Table 1 shows the proportion of participants without university degrees, with undergraduate degrees, and with postgraduate degrees. The average age was 30 years (σ = 7.15) and 33 years (σ = 13) for male and female participants, respectively. Recruitment was conducted online. Participants were compensated

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2940669, IEEE Access

TABLE 1. PARTICIPANTS' DEMOGRAPHIC FOR A USER STUDY

Education Level	Percentage
Non-graduate	24.75%
Undergraduate	48.5%
Postgraduate	26.75%

for their time and they had to meet the following criteria to be eligible:

- Own either an Android- or an iOS-based smartphone.
- Have experience with basic smartphone operations.

The survey was administered online and consisted of several questions, both multiple-choice and open-ended. The first part of the survey collected participants' demographic details including education level, age and gender, and the second part asked questions on cybersecurity knowledge (e.g. "how much do you know about information security?", "how and when did you learn about it?", "are software updates important for your device?"), cybersecurity behaviours (e.g. "do you update your device when prompted?", "do you open link from an unknown email sender?", "how often do you read internet policies, terms and conditions?") and privacy concerns and preferences (e.g. "do you lock your smartphone?", "what are the attributes of a good password?", "would you like to receive security tips and advice for your device?").

B. USER STUDY RESULTS

Cybersecurity Knowledge and Behaviours: Participants aged between 45-54 were most confident with their knowledge level (38.46% reported high level and 61% reported low) whereas participants aged between 18-24 were least confident (11.9% reported high and 88.1% reported low). Female and male participants had about the same level of confidence with 19.15% and 20.50% respectively reporting high. Overall 22.8% of participants felt they had low cybersecurity knowledge while 59.22% felt they had just enough knowledge to protect themselves online and 18% had reported themselves as highly knowledgeable.

The objective assessment of actual knowledge revealed that these assessments were highly inaccurate (Table 2).

An alarming number of the participants did have very little cybersecurity knowledge as evidenced by the fact that 59.71% (n=123) being likely or very likely to open links from unknown emails, which is the common means of launching a phishing attack. This means a vast majority of the participants are susceptible to phishing attacks (p<0.001, Student's t-test) [40]. In a one-way multivariate analysis of variance with both Welch's [41] and Hotelling Lawley t-test [42], we found that education levels have no significant impact on security-conscious behaviours.

Despite the low cybersecurity knowledge levels and insecure behaviours, participants were strongly motivated to protect their private data and acted to do so. The most prevalent security behaviour was the use of screen lock authentication on smartphones (n=165, 80%).

Privacy Concerns and Preferences: The majority of participants confirmed that protecting personal privacy was very important (n=155, 75%) and that they would benefit more if they received security advice and tips on how to protect their devices.

IV. ESTIMATION OF USER MENTAL EFFORT ON A PHISHING PAGE

This section outlines the second part of the study - estimation of mental effort displayed when viewing websites on smart devices, and the implications of mental effort level on phishing website identification. We present the tools used for the experiment, the procedure and the results.

A. STUDY TOOLS

The study used the following tools:

- JINS MEME smart glasses
- Android smartphone (Nexus 7) with a phishing game
- Smartphone for recording JINS MEME readings

JINS MEME is an eyewear that can detect head and eye movements [9]. The prototype looks just like normal eye glasses and it is made of 3 electrodes (electrooculography) and motion sensors (accelerometer and gyroscope) around the nose to detect eye and head movements respectively, as well as a Bluetooth LE module to stream the data to a computer, smartphone or tablet. JINS MEME smart glasses measure EOG signals of roughly over 100 *Hz* generated by human brain non-invasively and over 50 *Hz* of the motion sensor. Additionally, JINS MEME provides raw data that could easily be visualized and retrieved in CSV. The smart glasses battery can operate up to 8 hours [13].

We developed a simple Android quiz game for participants with three main interfaces. The start screen shows a start button, a score line and the page number. The second screen contains 20 different pages which are a mix of emails and webpages. The pages include both legitimate and phishing samples and the user has to select which sample (phishing or legitimate) is being shown on each page. The last screen shows the player's score at the end of the quiz. We twisted the webpage and email features such as URL, logo, graphic, structure to look as close and authentic to legitimate ones so that it is difficult to tell the difference. Thus, to get a good score, a user needs a certain level of vigilance to identify a phishing web page. Fig. 1 shows the sample interfaces from the game. We sampled the phishing emails and websites from *https://www.phishing.org/* and *https://www.phishtank.com/*. Their contents are already verified as phishing by the respective organizations. For the legitimate ones, we collected random legitimate emails and websites.

B. EXPERIMENT SETUP

We sampled 40 participants to participate in the study, 50% computer science students and 50% non-computer science students. Table 3 shows the education level and IT expertise of the participants.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2940669, IEEE Access

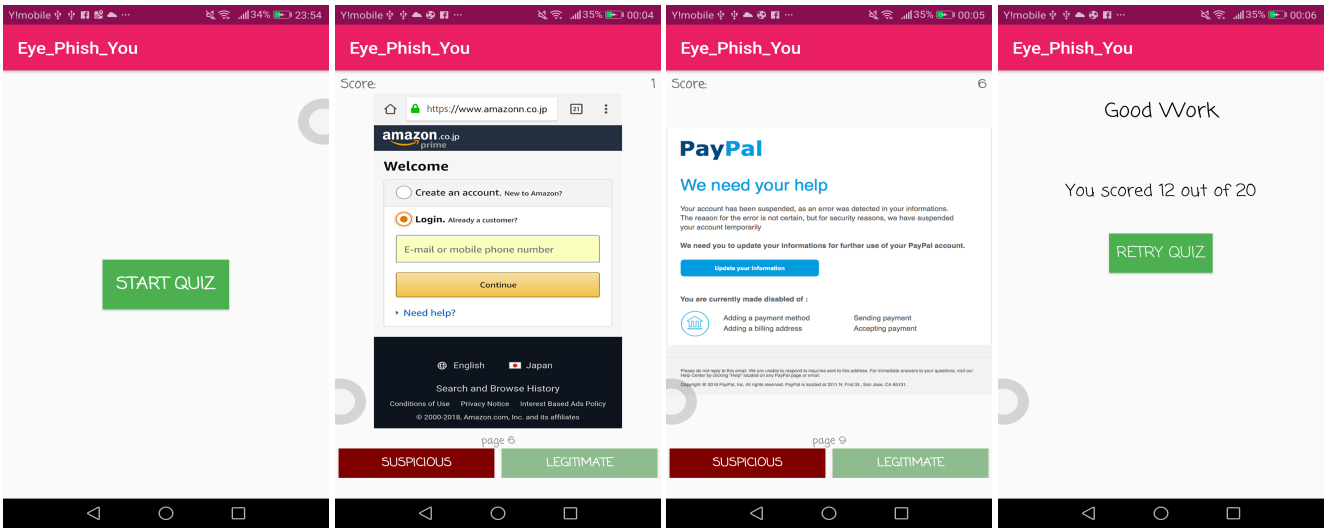


FIGURE 1. Screen shots of the game showing the starting screen, sample webpage/email and a user's score.

TABLE 2. PHISHING SUSCEPTIBILITY BASED ON OUR ASSESSMENTS

Demographic	Category	Susceptibility by (%)
Age	18 - 24	57.14%, n=40/70
	25 - 34	63.77%, n=44/69
	35 - 44	68.42%, n=26/38
	45 - 54	44.83%, n=13/29
	Average	*59.70%, n=123/206*
Education	Non graduate	66.67%, n=34/51
	Undergraduate	57.00%, n=57/100
	Postgraduate	65.45%, n=36/55
	Average	66.65%, n=127/206
Gender	Female	58.97%, n=46/78
	Male	60.15%, n=77/128
	Total Average	59.71%, n=123/206

TABLE 3. PARTICIPANTS' EDUCATION AND EXPERTISE FOR THE JINS MEME EXPERIMENT

Demographic	Category
Education	Undergraduate: 27.5%
	Postgraduate: 72.5%
IT Expertise	Tech-savvy: 50%
	Non-tech-savvy: 50%

The procedure was as follows: JINS MEME is turned on and connected via Bluetooth to a smartphone that has the official application for recording the signals. A user puts on JINS MEME glasses with the sensors intact to the nose pad and bridge. When the user is ready, she gives a signal and start the quiz. The starting of the quiz is synchronized with the start of the smart glasses recording by its application. Once a user completes the quiz, the recording also synchronously stops. The smart glasses use its own designated algorithm to calculate and estimate the mental effort of a user by checking frequently gazed points, starred points, deep focuses, normal focuses, eye movements, blink power and when a user is not focusing at all. Fig. 2 shows the JINS MEME smart glasses worn by one of the participants and Fig. 3 shows the varying levels of focus exhibited during the experiment.

FIGURE 2. JINS MEME smart glasses worn by one of the participants.

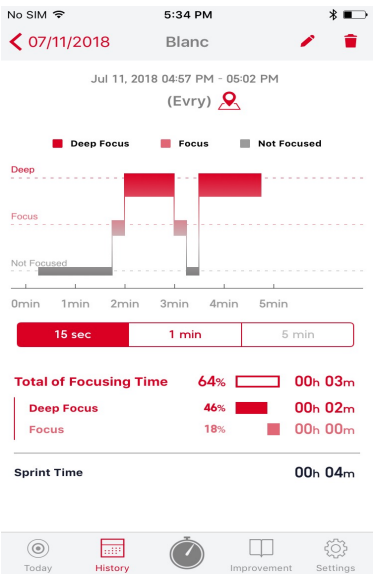


FIGURE 3. Sample signal recordings by the JINS MEME official App showing a participant's focus rate while taking up a quiz.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2940669, IEEE Access

C. JINS MEME EXPERIMENT RESULTS

We observed a strong positive correlation between the mental effort measured and quiz scores among tech-savvy users and inconsistent correlation among non-tech-savvy users. Therefore, tech-savvy users with higher mental effort had a higher phishing identification rate than those with lower or average effort. We calculated the correlation between the mental effort and quiz score for the tech-savvy users by using Pearson's product correlation [43] as shown in Table 4 and Fig. 4. The calculated correlation coefficient (r) is given by the Equation 1 and Equation 2. Whereas, the r_{xy} is the measure of linear dependence or association between x (mental effort) and y (phishing quiz score), and it ranges between -1 and 1.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (1)$$

$$r = r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (2)$$

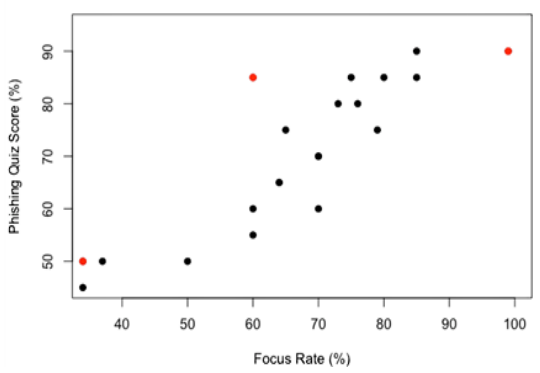


FIGURE 4. Correlation between mental focus and quiz score for tech-savvy participants.

For the non-tech-savvy participants, we noticed two things: some participants with extremely higher mental effort did not necessarily get higher quiz scores in the game and some participants with lower mental effort did not necessarily got lower quiz scores. We believe that the participants who had lower quiz scores despite higher mental effort did not have even a slight idea of how the phishing sites look. On the other hand, the participants who got higher scores despite exhibiting less mental effort may have just guessed the correct answers or had a great knowledge of the subject. However, on average, non-tech-savvy participants who exhibited reasonable mental effort performed as well as much as most of the tech-savvy participants. Fig. 5 shows the mental effort of non-tech-savvy users against phishing quiz scores and its statistical summary on Table 5. Red dots denote the odd outcomes i.e. high mental efforts with lower scores and vice versa.

Fig. 6 shows the overall correlation among tech-savvy and non-tech-savvy participants combined where the correlation

TABLE 4. A SUMMARY STATISTICAL TABLE FOR FIGURE 4

t	df	p-value	confidence interval		correlation
7.0281	18	1.472e-06	0.6658	0.9418	0.8561035

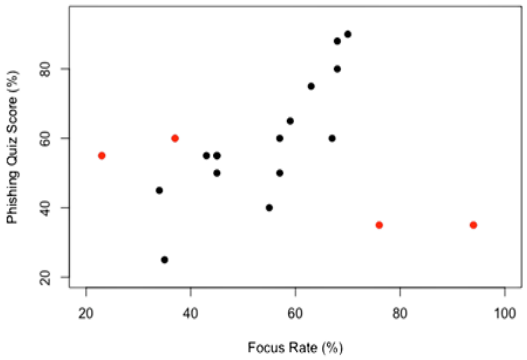


FIGURE 5. Correlation between mental focus and quiz score for non-tech-savvy participants.

coefficient (r) is 0.4882285 (about average) and data significance with p-value = 0.001393.

D. POST-EXPERIMENT INTERVIEW

After every game session, we held a brief interview about the participants' experience. We also played the game with each participant in order to explain the answers. All participants agreed that the game and the scores they archived were fair. However, we noticed that a good number (about 38%) of participants, even among the tech savvy, did not know some of the phishing indicators. Some of the missed indicators were just due to lack of attention and some were due to lack of knowledge or experience with phishing attacks. Most notably was the *https://www.amazonn.co.jp/* where most participants missed the additional '*n*' because naturally, the text was not big enough due to the screen size, and roughly, it was easy to think that the URL was legit. Additionally, some of the participants had no idea that if you are addressed as "*Dear user, Dear friend, Dear customer* etc." could be one of the biggest hints that you are on a phishing email. Legitimate businesses address their customers with their names but phishers do not often know the identity of their targets.

V. DISCUSSION

In this section, we discuss the role of attention and prior phishing knowledge on the ability of a user to identify phishing attacks. We also discuss UnPhishMe application prototype and its improvements for assisting users with phishing attacks identification.

TABLE 5. A SUMMARY STATISTICAL TABLE FOR FIGURE 5

t	df	p-value	confidence interval		correlation
0.9811	18	0.3846	-0.2607	0.5940	0.2055

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2940669, IEEE Access

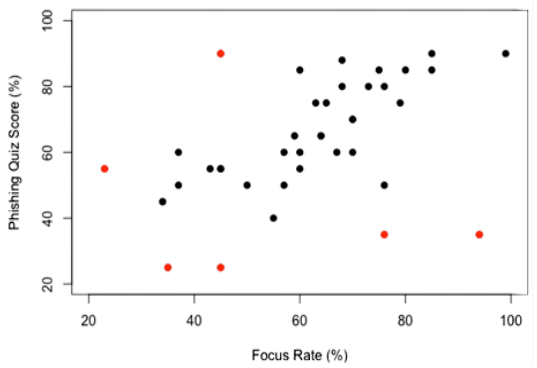


FIGURE 6. Correlation between mental focus and quiz score for all participants.

A. FACTORS THAT CAN SUPPORT PHISHING SITE IDENTIFICATION

The results derived from our study with JINS MEME eyewear suggest that the combination of good computer literacy and reasonable attention to a webpage has a positive impact in identifying phishing attacks. However, little focus can easily lead a user into skipping important cues for phishing attacks regardless of phishing knowledge and awareness as shown in Fig. 4. The performance of non-tech-savvy participants (Fig. 5) and for all users (Fig. 6) also suggests that a lot of effort does not guarantee a higher chance of identifying phishing attacks either. Moderate and reasonable effort can yield better phishing identification chance in both cases.

Our study, like few others [17] [18], indicates that being aware of phishing attacks alone is not enough for a user to stay safe from phishing attacks. The awareness of risks has to be linked to a perceived vulnerability or a mitigation strategy. *Downs* et al. found that prior experience and knowledge of phishing attacks architecture help to predict behavioural responses to phishing attacks [19], but this is not contrary to our findings or those from the studies in [17] [18] because experience with phishing attacks and architecture is very different from mere awareness about existence of the attacks. The study by *Downs* et al. further suggests that deeper understanding of the web architecture, such as being able to precisely interpret URLs, page redirection and understanding other cues such as what a lock signifies, is associated with less vulnerability to phishing attacks. It also posits that perceived severity of the consequences does not predict secure behaviour. Thus, efforts to educate users on the architectures and security features on browsers, rather than merely informing them on what phishing is and warning them about the risks is necessary. Many universities now require students to read policies related to IT facility usage before being granted access to use them, and these policies usually have sections on cybersecurity [20] [21]. In spite of this, our study showed that even people with higher education level do not necessarily have higher cybersecurity knowledge. Previous studies have shown that policy statements are rarely read [22] [23], and therefore more engaging means to educate users should be employed.

Additionally, *Chen* et al. posit that behavioral characteristics such as intolerance of risk, curiosity, and trust can be used to predict individual ability to identify phishing interfaces. In "*Real or Bogus*", they investigated characteristics of users and how they influence their ability to predict phishing attacks. They found that participants who are intolerant of risk were more likely to regard legitimate interfaces as phishing. In contrast, participants who were more trusting and less curious performed better on a phishing security quiz [24]. In our study, we found that women, who are usually more risk-averse in cyber and other behaviours than men [25] [26], were only less susceptible if they had higher education levels. This might be because higher education level means they are generally more computer-literate, and therefore understand or have had experiences with cyber-attacks such as computer viruses or spam messages. Statistically, our results show that the females whose education levels are at least bachelor degrees are less susceptible to phishing than men of the same education levels (21% vs. 38% respectively exhibited insecure behaviours). On the other hand, females with no bachelor degrees are more susceptible than men in the similar category (31% vs. 12.70%, respectively).

B. RECOMMENDED SYSTEM STRATEGY FOR PHISHING DETECTION ON SMARTPHONES

Based on our conducted studies and experiments on smartphone behaviours of users and the estimation of their mental efforts on identifying phishing attacks, we reckon that users require automated assistance for phishing attacks. Users are responsible for understanding how phishing attacks work and make efforts to learn about it through various awareness programs. However, leaving the entire burden of phishing identification to users may not be a plausible idea as we have seen that awareness alone is not enough. For instance, when a user is tired or exhausted she can make irrational decision despite being aware of the phishing attacks. Thus, we recommend an anti-phishing strategy that can independently and automatically detect phishing attacks and intervenes during the user login process. The key focus of the strategy is low-computational power requirement. As we have already discussed in Section II.A, most existing anti-phishing strategies are not suitable for small-sized devices such as smartphones.

In our previous work [34], we developed a prototype of an Android application (UnPhishMe) that simulates a user login procedure by using dummy login credentials to thwart phishing attacks. We used the automation framework (Selendroid [27]) which has the ability to manipulate the User Interface (UI) of Android native and web apps. In this section, we outline the system design, performance analysis and the weakness of the prototype and how it can be improved and implemented as a native Android browser.

1) OVERVIEW

UnPhishMe's logic is to automatically intervene in the login page in the background process of a device. It simulates user

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2940669, IEEE Access

authentication procedure through lightweight Java classes and methods. It intercepts a login page opened by a user and simulates the login procedure with fake credentials. Technically, an authentication attempt to a login webpage with incorrect login details examines the trustworthiness of that page. However, a user needs to have prior knowledge and remembers to do so every time she encounters a suspicious page. In small size devices, this procedure could be very tedious when done manually.

Our proposed approach uses Java lightweight library class (HTTP request client) that incurs a very small amount of CPU and memory to analyze a destination server HTTP header information [45]. In addition, instead of analyzing the URL as a string of characters for determining its changing dynamic, we analyzed it as numeric characters by programmatically computing its hashcode every time a page was loaded. Comparing numerals is much faster and easier than comparing strings.

2) SYSTEM DESIGN

Normally, when a user successfully authenticates herself to a legitimate webpage, she expects to see a new webpage with some requested resources. Technically, the current URL string changes to a new or extended URL string.

This is also true for a phishing webpage, however unlike a legitimate webpage, the login inputs correctness do not matter. Contrary to that, on a legitimate website if a user provides incorrect login inputs there will be various outcomes. Certainly, the login inputs will be rejected and the webpage remains the same with its contents slightly altered *e.g.*, a display of an error message. As a result, the URL string remains unaltered and the page does not shift. However, some websites such as Facebook [28] provide more than one alternative login pages. Once a first login attempt fails, a user is presented with another similar page, which is modified with some additional input requirements such as reCAPTCHA. A page shift automatically alters the URL string into a new one. Therefore, its new computed hashcode will be different from its original hashcode even if the authentication fails. Thus, the login automation should be iterative in such case as summarized in the system overview in Fig. 8.

By actively monitoring the URL changes after the authentication attempt, the application can compare the original URL hashcode h(URL) and the subsequent URL hashcode h(URL'). In principal, the URL exhibited before a user logs into a website remains the same even after the authentication attempt fails on a legitimate webpage. If a user is successfully authenticated, it is certainly expected that the URL will change into a new one. For a malicious site, the URL almost always changes with a successful authentication even when a user provides incorrect login credentials [44].

UnPhishMe consists of two engines, one is for an authentication automation and the other one is for server response interpretation and alert generation to a mobile device user as shown in Fig. 7.

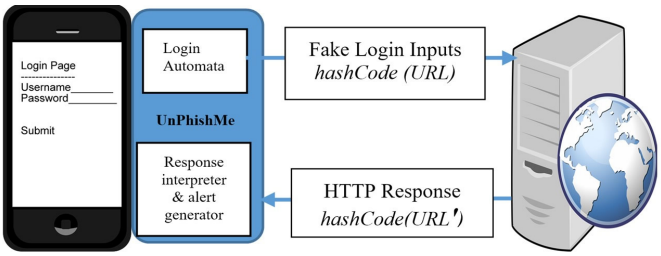


FIGURE 7. Checking HTTP response and URL consistency.

The login automation module sends login credentials to a resource server and computes a current URL hashcode. Later the resource server response about the authentication is given and again the new URL hashcode h(URL') is computed for comparison with the original one. Thereafter, the server response and the URL string status are intercepted by the application to generate a proper warning message for the user.

Algorithm 1 IsClientValidate(userID, password, Flag)

Result: true or false
initialization

```
if Flag == 0 then
    if serverConnection==0  userID.contain(@) then
        return true
    else
        return false
    end
else
    if serverConnection==0 then
        return true
    else
        return false
    end
end
```

In brief, we developed algorithms 1 and 2 (reference to pseudocode). Algorithm 1 handles client-side validation issues on webpages *e.g.*, checking whether the username is an email or has the minimum length of characters, while Algorithm 2 monitors the authentication response of the destination server. In our experiments with Alexa websites, we found that most websites, 198/200 (98%), did not have hard rules for usernames *e.g.*, types of characters used. They only specified the minimum length. We therefore created a database of usernames that satisfy the different lengths specified and email addresses using these usernames.

When a webpage loads, Algorithm 2 invokes Algorithm 1. Initially, Flag is set to 1 in Algorithm 2. This means the algorithm enters a username, not an email address. The algorithm then enters and attempts to send a username from our database and a password. If the destination server requires an email address, the client-side validation script on

the webpage will not allow these details to be sent. In this case, Algorithm 1 return true where there is a zero connection to the destination server (*serverConnection*=0) and sets the Flag to 0 in Algorithm 2 to denote an email address is needed. Setting the Flag to 0 allows the algorithm to enter and send an email address from our database along with a password. In both cases, if Algorithm 1 returns false (*serverConnection*=1), the entered login details have successfully been sent for authentication checking and therefore Algorithm 2 takes over to monitor the appropriate response.

Algorithm 2 TestPhishingSite()

```
Result: Alert User
initialization
  URL or Link
  Count=10

start loginPage()
  Flag=1
  if !(IsClientValidate(userID, password, Flag)) then
      if (!IsAuthenticated(userID, password)) then
          if hashcode1=hashcode2 then
              if http_status_code is in {400,499} then
                  Legitimate Site
              else
                  other error
              end
          else
              while Count!=0 do
                  TestPhishingSite()
                   Count-=1
                   Flag=0
              end
          end
      else
          Suspicious site
      end
  else
  end
  return Legitimate site
```

The type of operations for the algorithms is elementary, because all the operations are checking and affectation. Thus the time complexity is linear i.e.,

$$T(TestPhishingSite()) \in \theta(n) \tag{3}$$

As for the space complexity, an array for error codes needs 99 units (400-499) and all other remaining variables require just 1 unit of space. That means, the space complexity is also linear such that.

$$DSpace(TestPhishingSite()) \in \theta(n) \tag{4}$$

3) EXPERIMENT AND RESULTS

We show the experiment procedure and results of our application in this section. We depict its detection accuracy, significance and computational cost.

We tested about 700 websites, 500 suspicious ones from Phishtank [14] and a supplementary test with top 200 legitimate ones from Alexa.com [35]. Most of the suspicious websites (96%) responded positively to the authentication automation with completely dummy login credentials. A positive response, a successful authentication and inconsistent URL string, raised an alert every time a malicious site was opened. In the supplementary test with 200 legitimate websites, we obtained an overall accuracy of 91%. We explain these results in the V.D.(2) section of the Discussion.

We evaluated the performance of UnPhishMe through an Android device with the following specifications:

- Type: Samsung Galaxy SIII
- Android version: 4.2.2
- Mode: GT-I9300
- CPU: ARMv7 Processor, 1400 MHz, 4 Cores
- Internal memory: 853 Mb
- Internal storage: 11,000 Mb

Memory usage and performance differs from one device to the other. The lower limit for a low and medium density screen device is 16 Megabytes (Mb). That is a baseline for Android memory usage by an application [29]. We have optimized UnPhishMe CPU performance and memory usage to a minimum by implementing lightweight Java classes and methods. The measurements of CPU and memory usage while using UnPhishMe were less than 5% and 10Mb respectively, which is relatively low for a mobile device. We show the performance test results of UnPhishMe from an Android Profiler tool of the Android Studio [30] in Fig. 9.

4) SCOPE AND ASSUMPTIONS

This experiment had several assumptions and a scope that is described in this section. We discuss some implementation and performance issues and provide some recommendations on how to improve this work.

Our work focuses on mobile devices that use an Android operating system. However, it is not limited to Android-based devices only, it can be re-developed for other device operating systems such as iOS (formerly iPhone OS). However, currently, due to iOS restrictions to manipulate its drivers, we could not implement our approach on it. Furthermore, since our solution depends on RFC 2616 [31] industrial standard response for client requests, we are limited to legitimate websites that conform to that standard. Some web servers, such as *http://www.nike.com*, that implement such server standards for an authentication failure, fit well into our solution scope as indicated in Fig. 10. Together with other criteria, it is easier to deduce a phishing site when it does not behave like a legitimate one. The standard information can be a certain response code equivalent to a client request such as HTTP 401 Unauthorized or 403 Forbidden.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2940669, IEEE Access

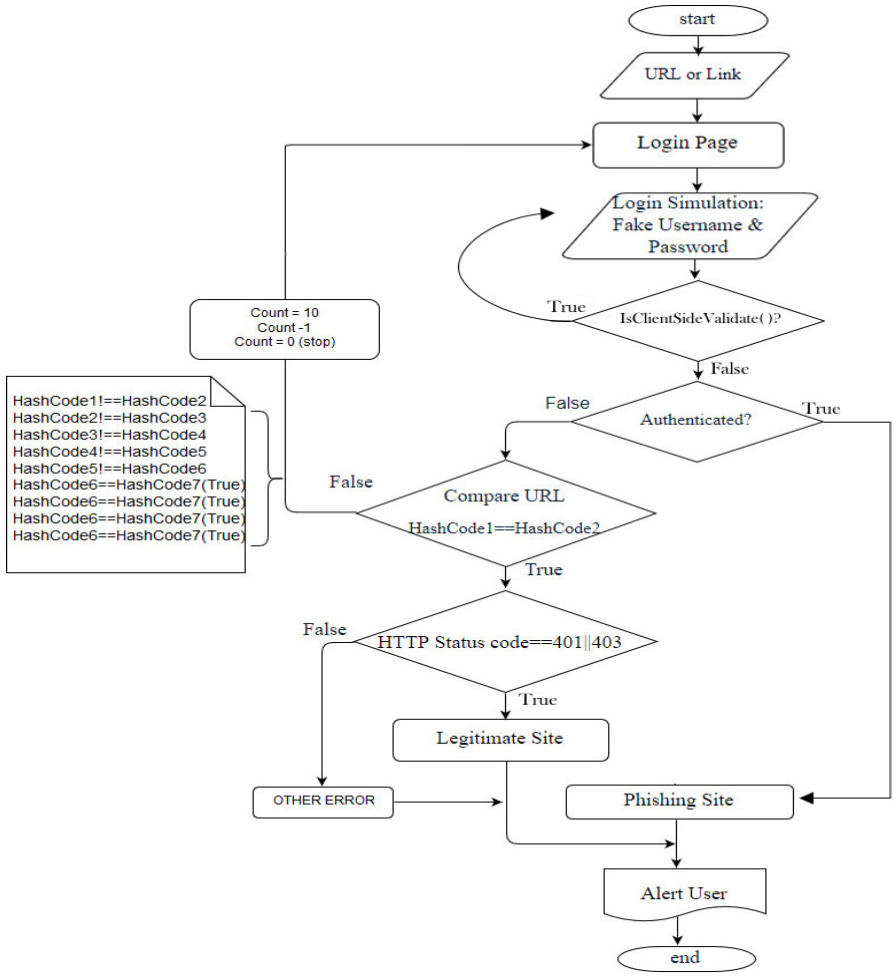


FIGURE 8. The work flow of UnPhishMe.

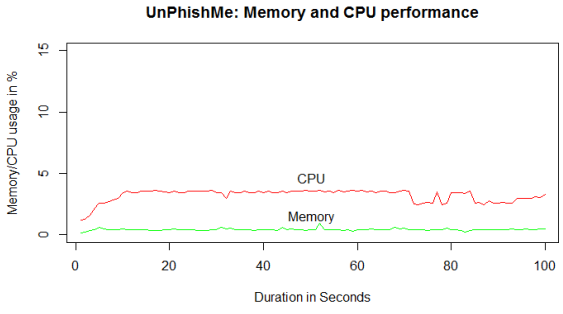


FIGURE 9. UnPhishMe application prototype CPU and memory usage.

C. RECOMMENDED SYSTEM IMPROVEMENTS

Despite the UnPhishMe prototype being suitable for a real-time phishing detection and timely feedback, it is not a replacement for a browser. However, the entire logic of the prototype can be implemented inside a native Android browser which by default does not offer any form of phishing protection [32] and can freely be customized. Moreover, this kind of implementation waives off the need of a browser to

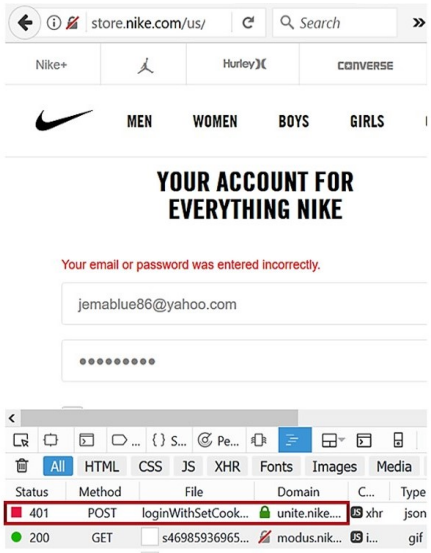


FIGURE 10. Implementation of the standard RFC.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2940669, IEEE Access

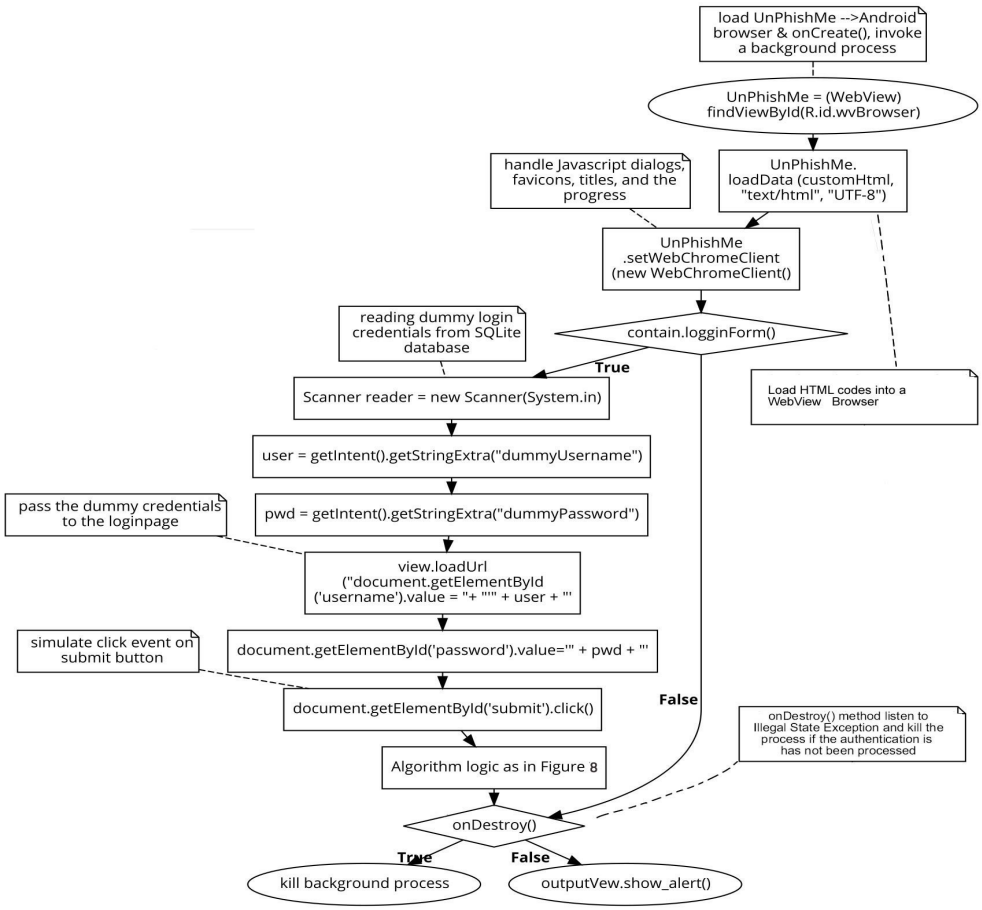


FIGURE 11. Implementation of UnPhishMe algorithm with an Android WebView browser.

have third party anti-phishing plugins which are computationally heavy.

1) IMPLEMENTATION ON A BROWSER

Without changing the logic of UnPhishMe prototype, we initiate and customize an Android browser by modifying a Web View (WV) through the Android WebKit and invoke the input data in the background process. For setting up the custom Web View browser, the WV object has to be created whereas, the *WebViewClient* handles *onPageFinished()*, *shouldOverrideUrlLoading()* methods, etc., which are responsible for controlling the web format and structure when loaded while the *WebChromeClient* handles Javascript's *alert()* and other functions.

When a webpage is opened by a user, the *shouldOverrideUrlLoading()* method is called and the browser checks for the existence of the HTML login form fields by scanning the HTML tags. If there are no login form fields available, the process is killed by setting the Boolean *onDestroy()* function to TRUE. To detect if the page contains a login form, our approach looks for the HTML form action attribute, *<form action = " " method = " " >*. If the login form fields exist, the browser invokes the retrieval of dummy credentials from an SQLite database which is automatically installed when the browser is installed. The *getElementById()* method is then used to inject the credentials to the appropriate login fields and the *onClick()* method submit the form. Meanwhile the boolean *onDestroy()* function is set to FALSE so as the process could be finalized and an alert generated to the user. We have summarized this procedure with a diagram in Fig. 11.

2) LOGIN CREDENTIALS GENERATION AND MANAGEMENT

The fake passwords used by UnPhishMe were generated using a simple script (Listing 1). In our experiment with Alexa sites, we found that apart from one website, the top 200 sites follow the same password requirements. The exceptional website did not accept '@' or '-' as special characters. Listing 1 generated a random list of characters that satisfy the password requirements of the top 200 websites. We used this approach to ensure that if a *Phisher* implements a client-side validation for a password, the fake password would be viewed as legitimate since it meets the required criteria. However, in our testing of phishing sites, we only found rare cases of input validation on userID (either an email or a username is acceptable) but not on entered passwords. The fake passwords were stored in HashMap tables of an SQLite

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2940669, IEEE Access

database through an Android dbhelper.

Listing 1. Password Generator Function

```
function pass_gen (len) {
var length = (len)?(len):(10);
var string='abcdefghijklmnopqrstuvwxyz';
var numeric = '0123456789';
var pct='!@#%^&*()_+~`|}{[]\:;?><,./-=';
var password = "";
var character = ""; var crunch = true;
while(password.length<length) {
unphish1 = Math.ceil(string.length*Math.
    ↪ random()*Math.random());
unphish2 = Math.ceil(numeric.length*Math.
    ↪ random()*Math.random());
unphish3 = Math.ceil(pct.length*Math.random
    ↪ ()*Math.random());
up = string.charAt(unphish1);
up = (password.length\%2==0)?(up.toUpperCase
    ↪ ()):(up);
character += up;
character += numeric.charAt(unphish2);
character += pct.charAt(unphish3);
password = character; }
password=password.split('').sort(function(){
return 0.5-Math.random()}).join('');
return password.substr(0,len); }
console.log( pass_gen () );
```

3) BENEFITS OF THE RECOMMENDED IMPROVED IMPLEMENTATION

An implementation of UnPhishMe's logic on a web browser reduces the risk of a user exposing her login information to an attacker. Most importantly, there is no need for a web browser to have third party plugins which normally slow down mobile devices with limited computational power. Additionally, this implementation reduces even the risks that are not associated with an action of a user. For instance, phishing attacks that are associated with DNS and Web cache poisoning or typo squatting. Thus, if an attacker compromises a resource, proxy or DNS server, and redirects the user's traffic to a rogue server, the user can still be safe with the assistance of UnPhishMe.

We obtained the performance data dump from the Android Studio 3.1.3 Profiler. With the same testing environment, UnPhishMe implementation on a browser had about the same amount of memory usage as the prototype, which ranges between (3% - 4%) of unused memory (500 Mb) which equates to a range of 15 - 20 Mb. However the CPU usage was about 25% which is 20% higher than the prototype as indicated in Fig. 12. Additionally, the average network usage was 3.5 Mbps.

In future work, this automatic functionality can be fully integrated into commercial smartphone browsers with full browser functionalities. Furthermore, since the method used in our application is constantly scanning the HTML tags for login fields, we intend to improve it so that it can detect the presence of login fields without scanning the entire list of HTML tags.

In comparison with similar approaches in iOS [36], [37], [38], [39] and PC, our approach fares much better as it eliminates zero-day or zero-hour phishing attacks, with limited resources possible, in a real-time environment. As we mentioned earlier, our design is simple thus it favors small-sized devices computationally but does not favor large-sized devices. The complexities of the algorithms are demonstrated in Equation (3) and Equation (4).

In iOS, however, the restrictions necessitate most researchers to test their algorithms in Android-based devices, which have less

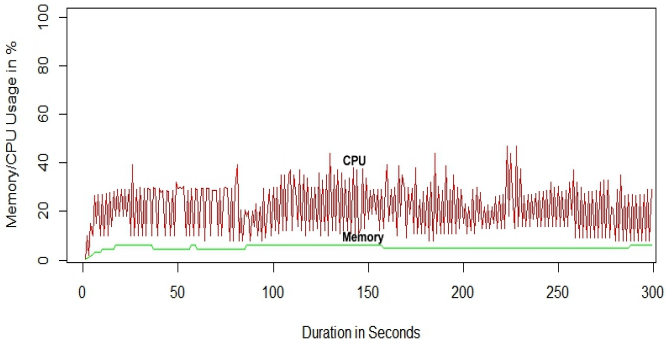


FIGURE 12. Performance evaluation of UnPhishMe on a Webview browser.

restrictions. The iOS architecture is closed-source thus its kernel restricts the developers to have a full control over the hardware, timing, file system, interrupts, drivers and power management. As a result, in iOS, implementing detective techniques like ours is very difficult unless a developer illegally manipulates the jail-broken iOS device to get a root access. Thus, we could not find this technique implemented in iOS. However, there a few solutions that work on iOS. Again the main challenge is that they have similar disadvantages that motivated us into developing our solution. For instance, *Anti-Phishing - Identity Guard* [36], *Metacert anti-phish app* [37], *Avira Mobile Security for iOS* [38], and *Defense against mobile phishing attack* are some of the anti-phishing techniques that are available in iOS [39]. However, they work by collecting information about known phishing sites from many sources and automatically update their list of phishing sites. Thus, they are not different from any other blacklist-based techniques which are not able to efficiently perform in a real-time and on zero-day or zero-hour phishing attack. Moreover, applications such as *Avira mobile solution for iOS*, for instance, necessitate the use of cloud or proxy server to separately analyze the heuristic features for phishing detection. For example, we tested some of these applications with malicious websites from Phishtank, they took at least a couple of seconds up to two minutes to determine whether a simple and obvious phishing site is malicious. In that case, a user might divulge her sensitive information even before an alert is generated.

D. LIMITATIONS AND FUTURE WORK

1) ARCHITECTURAL ISSUES

The authentication automation in mobile devices is more challenging than in desktop computers. The necessary automation drivers that are available on desktops are normally not found on mobile devices. In this work, we automated the user authentication procedure through Selendroid with JavaScript Object Notation (JSON). However, Selendroid is only compatible with Android devices and may not work on other types of devices.

Another significant challenge for implementing our solution is the requirement of specific websites that implement or follow industrial standards for authentication such as RFC 2616. Following such standards makes it difficult for an attacker to simulate fake legitimate responses. However, there is a possibility for an attacker to try to authenticate the UnPhishMe request traffic to an associated legitimate website on the fly in order to fool a user into providing true credentials. UnPhishMe has several numbers of iterations for addressing that problem, as we have demonstrated in the system model in section V.B.2. Additionally, a client-server connection time can be shortened on a client side to give an attacker less space to subvert a user request to a legitimate site. However, semantically,

an attacker always attempts to exploit the naivety of some website users rather than exploiting weaknesses of a system [33].

2) DETECTION ACCURACY

Despite attaining an overall detection accuracy of 96% for phishing sites from Phishtank, our approach missed about 4% of the suspicious entities. The failure was due to the non-standard HTML-based pages such as Flash- and PDF-based. Our approach depends heavily on the availability of HTML tags (input ID, name, type, form action) for login fields such as username, password and submit button. However, webpages are made of the HTML technology, often in conjunction with other technologies such as CSS, Javascript, JSON etc., thus non-HTML sites would render so poorly on smartphones to the point that they can easily be recognized effortlessly even without the techniques such as UnPhishMe.

On the other hand, another significant issue worth mentioning is the performance of our approach on legitimate webpages. We tested the approach with additional top 200 webpages from Alexa.com and achieved a 91% detection accuracy. The unsuccessful 9% were due to the webpages that require different authentication techniques such as QR code scanners, One Time Password (OTP) sent to a user mobile phone through SMS, and graphical passwords which work by having a user selects a matching image, in a specific order, presented in a graphical user interface (GUI). Since our approach is basically a text-based login automation bot, it cannot automate such complex authentication. However, technically speaking, *Phishers* may find it difficult or nearly impossible to replicate these kinds of authentications. Moreover, these kinds of websites can be white-listed in a module within UnPhishMe.

Lastly, only 7% of the tested legitimate webpages implemented client-side validation, they check whether the format of the input data is valid. However, this causes a small delay of 9-32 milliseconds. We believe this could be one of the reasons most legitimate websites (84%) did not implement client-side validation, rather they just check whether the provided credentials match their records in the back-end server.

VI. CONCLUSION

In this paper, we investigated the awareness of phishing attacks and cybersecurity behaviours of 206 users through a survey. We then used smart eyeglasses (electro-oculographic) to measure the mental effort and vigilance of 40 participants when browsing websites and when playing an Android phishing sites identification game that we developed. We found that knowledge and awareness of phishing attacks were insufficient for users' cyber protection because even knowledgeable participants had insecure behaviours like opening email attachments from unfamiliar senders. However, paying attention when browsing websites enabled participants to effectively identify phishing sites. Based on these results, we assert that browsers should automatically help users detect phishing websites as it is difficult to maintain high mental focus for long periods of time. This is particularly true for mobile device browsers, as the limited screens means it is difficult to see web browser security indicators and the limited capabilities mean many existing solutions are difficult to implement. We recommend automatic login with fake credentials to be deployed on browsers for phishing detection, as phishing sites tend to grant access even with wrong login credentials and this solution requires very low computational resources and would not require any user effort, making it sustainable and usable by both cybersecurity-aware and non-aware users.

REFERENCES

[1] R. Bitton, A. Finkelshtein, L. Sidi, R. Puzis, L. Rokach, and A. Shabtai, "Taxonomy of mobile users security awareness," Computers & Security, vol. 73, pp. 266-293, 2018.

[2] M. Jensen, A. Durcikova, and R. Wright, "Combating Phishing Attacks: A Knowledge Management Approach," Proceedings of the 50th Hawaii International Conference on System Sciences (2017), 2017.

[3] E. Kritzinger and S. V. Solms. 2013. A Framework for Cyber Security in Africa. Journal of Information Assurance & Cybersecurity (2013), 1-10. DOI:http://dx.doi.org/10.5171/2012.322399

[4] L. Wu, X. Du, and J. Wu, "Effective Defense Schemes for Phishing Attacks on Mobile Computing Platforms," IEEE Transactions on Vehicular Technology, vol. 65, no. 8, pp. 6678-6691, 2016.

[5] C. Yue and H. Wang., 2008, December. Anti-phishing in offense and defense. In Computer Security Applications Conference, 2008. ACSAC 2008. Annual (pp. 345-354). IEEE.

[6] M. Jakobbson, Understanding social engineering based scams. New York: Springer, 2016.

[7] H. Shahriar and M. Zulkernine, "PhishTester: Automatic Testing of Phishing Attacks," 2010 Fourth International Conference on Secure Software Integration and Reliability Improvement, 2010.

[8] L. Wu, X. Du, and J. Wu, "MobiFish: A lightweight anti-phishing scheme for mobile phones," 2014 23rd International Conference on Computer Communication and Networks (ICCCN), 2014.

[9] S. Kanoh, S. Ichi-Nohe, S. Shioya, K. Inoue, and R. Kawashima, "Development of an eyewear to measure eye and body movements," 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015.

[10] D. Miyamoto, G. Blanc, and Y. Kadobayashi, "Eye Can Tell: On the Correlation Between Eye Movement and Phishing Identification," Neural Information Processing Lecture Notes in Computer Science, pp. 223-232, 2015.

[11] D. E. Irwin, "Fixation Location and Fixation Duration as Indices of Cognitive Processing ," J. M. Henderson & F. Ferreira (Eds.), The interface of language, vision, and action: Eye movements and the visual world. New York, NY, US: Psychology Press., pp. 105-133, 2004.

[12] K. Kunze, M. Katsutoshi, Y. Uema, and M. Inami, "How much do you read?," Proceedings of the 6th Augmented Human International Conference on - AH 15, 2015.

[13] S. Ishimaru, K. Kunze, K. Tanaka, Y. Uema, K. Kise, and M. Inami, "Smart Eyewear for Interaction and Activity Recognition," Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA 15, 2015.

[14] W. Han, Y. Cao, and C. Lei, "Using a Smart Phone to Strengthen Password-Based Authentication," 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, 2011.

[15] H. Bridge, B. Goodger, G. Murphy, and J. N. Jitkoff , "Background auto-submit of login credentials." U.S. Patent 8,607,306.

[16] V. S. Bagad., and S. P. Kawachale. VLSI design. Technical Publications, 2008.

[17] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI 06, 2006.

[18] D. Harley and A. Lee, "Phish Phodder: is User Education Helping or Hindering?," in Virus Bulletin Conference Proceedings, 2007.

[19] J. S. Downs, M. Holbrook, and L. F. Cranor, "Behavioral response to phishing risk," Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit on - eCrime 07, 2007.

[20] N. F. Doherty, L. Anastasakis, and H. Fulford, "The information security policy unpacked: A critical study of the content of university policies," International Journal of Information Management, vol. 29, no. 6, pp. 449-457, 2009.

[21] A. Ghazvini, Z. Shukur, and Z. Hood, "Review of Information Security Policy based on Content Coverage and Online Presentation in Higher Education," International Journal of Advanced Computer Science and Applications, vol. 9, no. 8, 2018.

[22] A. D. Veiga, "Comparing the information security culture of employees who had read the information security policy and those who had not," Information and Computer Security, vol. 24, no. 2, pp. 139-151, 2016.

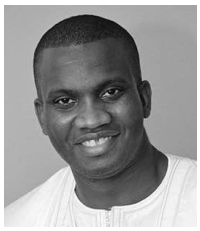
[23] "Cyber security - why you're doing it all wrong," ComputerWeekly.com. [Online]. Available: https://www.computerweekly.com/opinion/Cyber-security-why-youre-doing-it-all-wrong. [Accessed: 04-May-2019].

[24] Y. Chen, I. Yeckehzaare, and A. Zhang, "Real or bogus: Predicting susceptibility to phishing with economic experiments v1 (protocols.io.n74dhqw)," protocols.io, 2018.

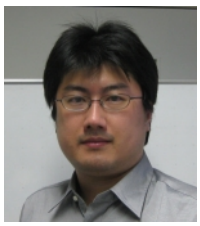
[25] S. Kleitman, M. K. H. Law, and J. Kay, "It's the deceiver and the receiver: Individual differences in phishing susceptibility and false positives with item profiling," Plos One, vol. 13, no. 10, 2018.
[26] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish?," Proceedings of the 28th international conference on Human factors in computing systems - CHI 10, 2010.
[27] M. Tan and P. Cheng, "Research and implementation of automated testing framework based on Android," Information Technology, p. 035, May 2016.
[28] Facebook. [Online]. Available: https://www.facebook.com/. [Accessed: 06-May-2019].
[29] https://developer.android.com/reference/android/app/ActivityManager.html getLargeMemoryClass (Accessed in May 2017).
[30] "Measure app performance with Android Profiler | Android Developers," Android Developers. [Online]. Available: https://developer.android.com/studio/profile/android-profiler. [Accessed: 07-March-2017].
[31] IETF. [Online]. Available: https://www.ietf.org/rfc/rfc2616.txt. [Accessed: 06-January-2017].
[32] N. Virvilis, N. Tsalis, A. Mylonas, and D. Gritzalis, "Mobile Devices: A Phisher's Paradise," Proceedings of the 11th International Conference on Security and Cryptography, 2014.
[33] B. Schneier, "Semantic attacks: The third wave of network attacks," Crypto-Gram Newsletter 14, 2000.
[34] J. D. Ndibwile, Y. Kadobayashi, and D. Fall, "UnPhishMe: Phishing Attack Detection by Deceptive Login Simulation through an Android Mobile App," 2017 12th Asia Joint Conference on Information Security (AsiaJCIS), 2017.
[35] Alexa top 1000. [Online]. Available: https://www.alexa.com. [Accessed: 26-June-2019].
[36] Anti Phishing - Identity Guard. [Online]. Available: https://apps.apple.com/us/app/anti-phishing-identity-guard/id1136376302. [Accessed: 26-June-2019].
[37] Innovative anti-phishing app comes to iPhones. [Online]. Available: https://www.computerworld.com/article/3327240/innovative-anti-phishing-app-comes-toiphones.html. [Accessed: 26-June-2019].
[38] Avira Mobile Security for iOS: Powerful protection against phishing attacks and identity theft. [Online]. Available: https://blog.avira.com/avira-mobile-security-for-ios/. [Accessed: 26-June-2019].
[39] J. Hou and Q. Yang, "Defense against mobile phishing attack," in Computer Security Course Project, http://wwwpersonal. umich. edu/yangqi/pivot/mobile phishing defense.pdf, 2012.
[40] Student, "The Probable Error of a Mean," Biometrika, vol. 6, no. 1, pp. 1-25, 1908.
[41] B. L. Welch, "The Generalization of "Students" Problem when Several Different Population Variances are Involved," Biometrika, vol. 34, no. 1/2, p. 28, 1947.
[42] C. J. Huberty, S. Olejnik, and C. J. Huberty, Applied MANOVA and discriminant analysis. Hoboken, NJ: Wiley-Interscience, 2006.
[43] K. Pearson, "Notes on the History of Correlation," Biometrika, vol. 13, no. 1, p. 25, 1920.
[44] L. F. Cranor, "Can Phishing Be Foiled?," Scientific American, vol. 299, no. 6, pp. 104-110, 2008.
[45] Unirest for Java. [Online]. Available: http://unirest.io/java.html. [Accessed in January 2017].

JEMA DAVID NDIBWILE received a Master of Technology (M.Tech) in Information Security from Jawaharlal Nehru Technological University (JNTU), Hyderabad, India in 2015. He is currently a Ph.D candidate at the Nara Institute of Science and Technology (NAIST) in Japan. His research interests include phishing countermeasures, the psychology of cybersecurity, and ethical hacking.

EDITH TALINA LUHANGA received a BEng in Electronic and Computer Engineering and an MSc in Advanced Computing Science from the University of Nottingham in 2010 and 2011 respectively and a Ph.D. in Information Science at the Nara Institute of Science and Technology (NAIST) in Japan. She is currently working as a Lecturer at the Nelson Mandela African Institution of Science and Technology (NM-AIST) in Tanzania. Her research interests include health behavior change, understanding users and human-computer interactions.



DOUDOU FALL received his M.E. degree in Data Transmission and Information Security from University Cheikh Anta Diop, Senegal in 2009. He received another M.E. & Ph.D. degrees in Information Science from Nara Institute of Science and Technology (NAIST), Japan in 2012 and 2015, respectively. He is currently an Assistant Professor in the Graduate School of Information Science, NAIST. His research interests include cloud computing security, vulnerability & security risk analysis.



DAISUKE MIYAMOTO is an Associate Professor in the Graduate School of Information Science and Technology, the University of Tokyo, Japan (UTokyo) since 2018. He received the Bachelor of Commerce from Kwansei Gakuin University, Japan, in 2000, the Master and Doctorate degree of Engineering from Nara Institute of Science and Technology, Japan (NAIST), in 2002 and 2009 respectively.

GREGORY BLANC received his Master's degree in Network and Information Security from Ecole superieure d'Informatique, Electronique et Automatique (ESIEA), France, in 2008 and a Ph.D. in Information Science from Nara Institute of Science and Technology (NAIST), Japan in 2012. He is currently an Associate Professor in Networks and Security at the department of Telecommunications Networks and Services at TELECOM SudParis. His research interests include intrusion detection, programmable networks, network virtualization, cognitive security, and IoT security.

YOUKI KADOBAYASHI received his Ph.D. degree in computer science from Osaka University, Japan, in 1997. He is currently a Professor at the Laboratory for Cyber Resilience, Nara Institute of Science and Technology, Japan. His research interests include cybersecurity, web security, and distributed systems. Dr. Kadobayashi is a member of ACM and IEEE Communications Society.