2017 IEEE Symposium on Security and Privacy

The Password Reset MitM Attack

Nethanel Gelernter
Cyberpion, Israel
College of Management Academic Studies, Israel

Senia Kalma, Bar Magnezi, Hen Porcilan
College of Management Academic Studies, Israel

***Abstract*—We present the password reset MitM (PRMitM) attack and show how it can be used to take over user accounts. The PRMitM attack exploits the similarity of the registration and password reset processes to launch a man in the middle (MitM) attack at the application level. The attacker initiates a password reset process with a website and forwards every challenge to the victim who either wishes to register in the attacking site or to access a particular resource on it.**

The attack has several variants, including exploitation of a password reset process that relies on the victim's mobile phone, using either SMS or phone call. We evaluated the PRMitM attacks on Google and Facebook users in several experiments, and found that their password reset process is vulnerable to the PRMitM attack. Other websites and some popular mobile applications are vulnerable as well.

Although solutions seem trivial in some cases, our experiments show that the straightforward solutions are not as effective as expected. We designed and evaluated two secure password reset processes and evaluated them on users of Google and Facebook. Our results indicate a significant improvement in the security.

Since millions of accounts are currently vulnerable to the PRMitM attack, we also present a list of recommendations for implementing and auditing the password reset process.

I. Introduction

A password is the primary and most popular mechanism for account protection. Users of web-services all use passwords to prevent unauthorized parties from accessing their accounts. For decades, this key role of passwords in the security world has attracted many hackers and security researchers.

The first computers had no need for passwords, and physical obstacles were the only security countermeasures. The need for passwords appeared with the rise of shared environments. Initially, passwords were saved in plain text. The first cases of password theft introduced the need for other solutions, such as using encryption, hashing, and salt [1].

Despite the improvements in secure password storage techniques, attackers still hack databases and get information about users and their hashed passwords [2]. The attackers then try to break the passwords offline using classical attacks like brute-force or dictionary attacks.

Even the most secure password storage will not help a user who chooses a weak password. Unfortunately, many users tend to choose easy to remember but also easy to guess passwords [3]. To prevent users from making this kind of mistake, many websites force their users to use strong passwords, or at least give them an indication about the strength of their password [4]. Enforcing strong passwords by applying restrictions to the user passwords and providing indications about the strength of the password were shown to be effective [5]–[8]. In addition to the strong password requirement, web-services such as banks, which allow sensitive operations, often force their clients to change their passwords frequently.

Choosing a strong password and ensuring it is securely stored are imperative to maintaining account security. However, these efforts are not worth much if the password reset process is vulnerable to attacks.

The fact that many users tend to forget their passwords has raised the need for password reset mechanisms. Paradoxically, the security requirements for choosing strong unique passwords and periodically replacing them, only makes password forgetting more common [9], [10]. Today, most of the websites with a password-based login system allow users to reset a lost password.

Password resetting is a challenging process. The website needs to ensure that the user can prove her identity without that password. Most websites rely on the email address of the victim, e.g., by sending a reset password link to the email address that was used to register the website account. However, this becomes much more challenging for the very important websites that provide the email services.

Websites that cannot reset passwords via email address, and websites that support cases in which the user lost access to a registered email account, offer alternative ways to reset the password. These websites use security questions or other communication channels such as mobile phone to authenticate the user before she receives the option to reset her password.

This paper shows that existing password reset processes in many popular websites are vulnerable to attacks by a weak attacker. In particular, we characterize, research, and evalute a new attack, which we call *password reset man-in-the-middle (PRMitM)*.

In a basic PRMitM attack, a user accessed the website of an attacker to get a resource, e.g., free software. The attacker requires the user to login for free in order to access the resource. During the registration process, or via other cross-site attacks, the attacker gets the email address of the victim. Then, on the server side, the attacker accesses the email service provider website and initiates a password reset process. The attacker forwards every challenge that he gets from the email service provider to the victim in the registration process. In the other direction, every "solution" that is typed by the victim in the registration process is forwarded to the email service provider. That way, the cross-site attacker is actually a man in the middle of a password reset process.

Some of the challenges the attacker may come up against

DOI 10.1109/SP.2017.9

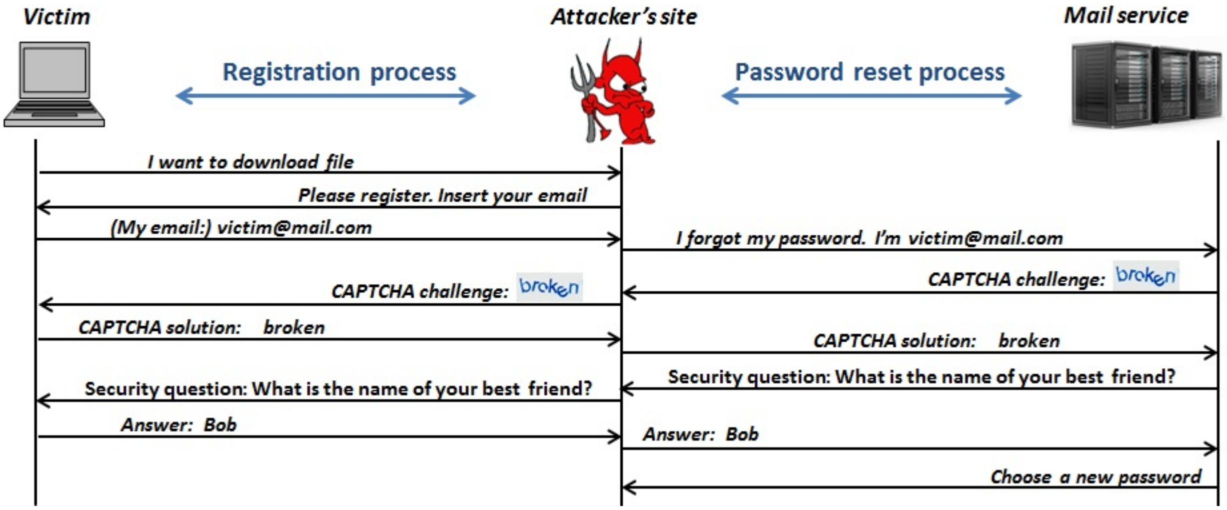
Victim
Attacker's site
Mail service
Registration process
Password reset process
I want to download file
Please register. Insert your email
(My email:) victim@mail.com
I forgot my password. I'm victim@mail.com
CAPTCHA challenge: broken
CAPTCHA challenge: broken
CAPTCHA solution: broken
CAPTCHA solution: broken
Security question: What is the name of your best friend?
Security question: What is the name of your best friend?
Answer: Bob
Answer: Bob
Choose a new password

Fig. 1: Basic PRMitM attack illustration. In this example, the email service provider challenges the attacker with a CATPCHA and a security question.

when he tries to reset a user's password are CAPTCHA challenges [11], security questions, and code that is sent to the mobile phone. Figure 1 illustrates a basic PRMitM attack.

Counterintuitively, websites that rely only on sending password reset message code to the user's mobile phone are sometimes more vulnerable to the attack. This is because the attacker can launch the PRMitM attack on them even in scenarios that are simpler than registration to a website.

We explore and analyze the different password reset SMS messages sent by popular websites to their users as well as password reset using phone calls.

We surveyed the password-reset mechanism of the most popular websites and of other popular email service providers, and analyzed how vulnerable they are. Our findings show that popular websites are vulnerable to PRMitM attacks, some of them very severely.

For example, we found that Google, the most popular website in the world, is extremely vulnerable to PRMitM attacks that exploit Google password reset using a phone call. We also evaluated the PRMitM attack using SMS messages on Facebook, the world's second most popular website. Beyond Google and Facebook, we found vulnerabilities in Yahoo!, LinkedIn, Yandex and other email services. We also discovered additional problems that occur in other websites and analyzed PRMitM vulnerabilities in mobile messaging applications like Whatsapp and Snapchat.

Beyond the surprisingly high number of vulnerable popular services, our findings include several problems, some of them surprising, that have not considered before in the design of secure password-reset process:

1) Informative password-reset messages do not prevent exploitation of users, mainly because many users ignore the text and just copy the code.
2) Users might be vulnerable to the attack, depending on their language settings. This is either due to difference in the content of password-reset messages in different languages or due to services that provide services in several languages, but send password-reset messages in another language.
3) The PRMitM attack can be used to take over accounts of very popular websites (e.g., Facebook) given minimal information about the user (e.g., phone number only). This allows easy exploitation in additional scenarios (not registration).

As existing designs of password-reset processes are vulnerable, we designed secure password reset processes using SMS and phone calls. We then evaluated their effectiveness on real Facebook and Google users with excellent results, mainly compared to the poor results achieved by their current mechanisms. We summarize our work with a list of recommendations for testing and improving the security of password reset processes in many websites.

A. Contributions

We make the following contributions:

1) Introduce the PRMitM attack, a new attack that exploits bad design of password-reset process in websites and applications.
2) Evaluate the PRMitM attack on Google and Facebook, the two most popular websites in the world.
3) Review the password reset processes of many popular websites and comparing the different approaches.
4) Explore further and identify similar vulnerabilities in popular mobile applications.
5) Design secure password reset processes using SMS and phone calls, and evaluate of them on Google and Facebook users. This was necessary, as our experiments indicated that in some cases, the straightforward solutions are not effective enough (see Experiment 2).

6) List recommendations for the secure design of the password reset process. Following the number of popular websites affected, this list is critical for quickly patching the vulnerabilities.

Our work has already helped several popular services improve the security of their password reset process. We believe it will help many other websites protect their users.

B. Organization

We begin with a description of the adversary model in Section II; this section also includes a survey that justifies the practicality of this model. In Section III, we describe the basic PRMitM attack. In Sections IV and V, we present and evaluate PRMitM attacks on password reset processes using SMS and phone-calls, respectively. Section VI shows that the PRMitM attack can also be launched on some mobile applications. Section VII presents possible defenses and evaluates them, and Section VIII discusses related work. The last two sections summarize our findings in a list of recommendations that can be used by websites to test and improve their password reset processes.

C. Ethics

Our institutes have no ethics committee. Nevertheless, we followed common sense and advice from experts to conduct the research ethically.

We reported our findings to the vulnerable vendors. Vendors that are severely vulnerable to the PRMitM attack, either fixed the vulnerability (Snapchat, Yahoo!) or informed us that they plan to fix the vulnerability (Google, LinkedIn and Yandex). Other websites, which are less vulnerable (e.g., Facebook) thanked us, and told us they will consider using our findings in the future, but they do not plan to apply fixes soon.

In the experiments we conducted, we avoided accessing information we did not get from the participants in advance. We also did not take over their accounts or change anything in their accounts. Additionally, we did not keep any private information beyond the final results (e.g., attack has succeeded or not).

D. Methodology Challenges and Limitations

This paper presents a set of attacks and evaluates them on different settings. Although the attack exploits vulnerability in the design of the password-reset process, the attack includes interaction with users. Hence, extensively rely on user studies and surveys. Totally, 536 participants took part in the surveys and the experiments that were done in this research; each of them participated only in once experiment or survey.

The need of many participants for both the surveys and the experiments was a technical challenge for us. Moreover, the nature of most of the experiments made this challenge becomes even harder. As our experiments simulate versions of the PRMitM attack, we preferred to rely on volunteers that will feel free to leave the experiment at any step. If participants get money, they might feel obligated to complete the experiment.

Like many other researches on related topics like phishing and password security, e.g., [10], [12], [13], we decided to rely on students from our institute. Although it is preferred to conduct larger user studies also on other populations, like other researchers, we believe that conducting all the experiments and the surveys with students gives good and reliable results that are relevant also for other populations. Other alternatives like Amazon Mechanical Turk workers (which is not available in our country) are not better, as there are many common characteristics to the users there.

Except of the ages of the students that were used to make sure that all the participants are adults, we did not collect any private information about the participants, as we did not think that this is necessary for the results. Of course, all the participants are required to be web users; otherwise, they cannot be used to evaluate the situations discussed in this paper. Like in most of the departments in our institute, the ages of the students in all the experiments ranged between 18 and 35, almost uniformly.

II. Adversary Model

To launch a PRMitM attack, the attacker only needs to control a website; no MitM or eavesdropping capabilities are required. The attacker attacks visitors of his website and takes over their accounts in other websites. This is similar to cross-site attacks like cross-site scripting [14], cross-site request forgery [15], and clickjacking [16]. We extend the discussion on the differences from cross-site attacks and from phishing in Section II-B.

In order to initiate the password reset process for a website in the name of the victim, the attacker needs basic pieces of information; these include items such as username, email, or phone number. This information can be extracted from the victim by the attacker during a registration process to the attacking website (Section III) or before some operations like file download, when the victim is required to identify herself using her phone.

For some websites, the attacker may be able to use cross-site attacks such as cross-site scripting [14], cross-site script inclusion [17], or newer techniques [18], [19] to gather details about the user. However, the use of these techniques implies restrictions, e.g., the user must be logged into the attacked website (see below for more details).

In addition to a visit to the attacker's website, the attacking page has to lure the victims into registering or inputting their phone number to get a code. To do that, the attacker can apply known and common methods. For example, the attacker can create a website that offers (or claims to offer) free services, e.g., streaming or files download. The website can require basic authentication (prove you are not a bot) before accessing some or all the services or to restrict them only for registered users. Section II-A shows that this requirement is reasonable.

A. Personal Details in Unknown Websites

Our attack is based on the assumption that users will agree to register or to have a one-time code sent to their phone in order to enjoy services online. Although it will be good for attacking website to provide valuable services to attract potential victims, in practice, the attacking website can only claim it is offering such services.

To test this assumption we conducted an anonymous survey among students in our institute. In the short survey, we asked participants whether they would agree to either register to a website or prove they are human using their phone or both the options, in order to use common online services such as file downloads for free.

Among 138 participants, only 6 claimed they will never register for unknown websites or give their phone number, no matter what free services are offered. Of the participants, 60.9% said they would agree to use both the options. An additional 27.5% would only agree to register, and the remaining 7.2% would only agree to identify themselves using their phone.

These results strengthen our assumption and show that the adversary model, in which victims register or authenticate themselves using their phones, reflects a common situation on the web.

Some of our colleagues were surprised by the willingness of users to use their phone number. For ethical reasons, we could not create a website with attractive content, and a fake website would not do the job. Hence, we conducted a simulation with the participation of another 99 students.

In this simulation, we described a website that stores files and requires a valid phone number to download them. The verification is done via SMS code, and the user is only required to insert his phone number.

We asked the participants whether they would agree to insert their phone number to receive the files in which they are interested. Of these, 39.4% said they would insert their phone number immediately, and 14.1% said they would first try to obtain the files via friends or via online SMS services. An additional 18.2% percent said they would insert their phone number only if they really needed the files (rather than just wanting them). In total, 71.7% of the participants would agree to insert their phone number.

B. *Comparison to Cross-Site Attacks and Phishing*

Visiting a malicious page might expose the user to several attacks. If the browser or one of its plugins has security bugs, an attacker could exploit these bugs to take over the entire machine. However, finding such bugs is considered a difficult task. Once a critical zero-day bug is discovered, it is quickly patched by popular browser vendors such as Chrome and Firefox.

Other risks come from vulnerabilities in the websites themselves, although it is challenging to find security bugs in popular websites. An attacker who wants to take over an account using classical web attacks like XSS [14] or CSRF [15], has to intensely explore each of its target websites. Without finding a vulnerability it is hard to know for sure whether the website is vulnerable or not. Unlike PRMitM, in cross-site attacks [14]–[16], [18] users must also be authenticated to the attacked website.

On the other hand, more interaction between the attacking page and the victim is required to launch PRMitM attacks. Unlike clickjacking and some XSS attacks, where only a few clicks are required, in PRMitM attacks, the victim is required to perform an operation in the attacking page and to insert at least a single minimal correct piece of information about herself, e.g., a phone number.

The need to insert private information is similar to phishing attacks in websites [13], [20]. However, in phishing attacks, the attacking page impersonates a legitimate website and tricks the victim into inserting her credentials (username and password). In PRMitM attacks, the victim is only required to give personal information (e.g., phone number) that users agree to give in order to get some services (see Section II-A).

Sophisticated phishing attacks might also follow similar application-level MitM approach to imitate legitimate websites or during the entire login process [21], [22]. Such a MitM approach might overcome also 2-factor authentication schemes, as the victim inserts codes and passwords into the phishing website. Hence, one might miss the most significant difference between phishing and PRMitM attacks: the vulnerability itself. Namely, for each of the attacks, there is a different answer to the question *what is being exploited?*

Phishing attacks exploit the users; there is no bug in the design of the attacked website and the attacker exploits unwary users who ignore indications given to them by the browsers. On the other hand, PRMitM attacks exploit bugs in the design of password-reset process.

The greatest challenge of the phishing attacker is the impersonation to another website. Users with minimal understanding can detect phishing attempts by carefully checking the site URL and whether HTTPS is on. Other anti-phishing solutions [23]–[26] make the launch of phishing attacks harder also against other users. The PRMitM attack obviates the need for impersonation; it can be launched naturally from every website.

As the PRMitM attack exploits server-side design bug, depending on the severity of the vulnerability, there is no chance for the users and other client-side defenses (e.g., browser built-in mechanisms or extensions) to detect the attack.

Table I summarizes the comparison.

III. MitM in Password Reset Process

This section describes the basic password reset MitM (PRMitM) attack, and presents the challenges and difficulties of the attacker. This section also surveys the mechanisms used by popular websites during the password recovery process.

A. *Password Reset MitM Attack*

The basic PRMitM attack exploits the similarity between the registration process and the password reset process. In both the processes, it is common to solve CAPTCHA challenges, answer security questions, get a confirmation link to the email, or to type in a code that is sent to a phone number. Hence, the attacker can take challenges from a password reset process of

	interaction with the victim	Login to the attacked website	Root cause (what is being exploited?)
PRMitM	Insert personal information	X	Bad password reset process design
Cross-site attacks	None or minimal (clicks)	√	Implementation bugs (usually)
Phishing	Insert credentials	X	The users themselves

TABLE I: Comparison to other attacks to take over accounts that require a visit in malicious website

a user, and present them to her as legitimate challenges during the registration process.

We now describe the attack in detail. For simplicity, we describe the attacked website as the email service provider of the victim. When a user initiates a registration process in the attacker's website, the attacker either asks the user to identify herself with her email address or launches another cross-site attack to extract it [14]–[18].

Once the attacker knows the victim's email address, he already knows both her email service provider and her username in this service. The attacker initiates a password reset procedure against the attacked website with the email address of the victim.

The attacker acts as man in the middle between the victim user and the attacked website in the password reset procedure. The attacker forwards almost every challenge (see Section III-C) from the attacked website to the victim under the cover of the registration process.

This process is illustrated in Figure 1. Given the email address of the victim, the attacker can similarly initiate a password reset process in the name of the victim in other websites, e.g., Facebook.

B. Challenges

We now discuss the four most common challenges that the attacker may encounter during the password reset process. The challenges are described from the easiest to the most difficult.

1) CAPTCHA Challenges: CAPTCHA challenges [11] do not aim to prevent an attacker from resetting the password, but rather aim to prevent the attacker from doing this *automatically*. A human attacker should be able to solve CAPTCHA challenges just like a human victim. However, to launch the PRMitM attack on a larger scale it is necessary to solve them automatically. Therefore, the PRMitM attacker forwards the CAPTCHA challenges to the victim users, and forwards the solutions submitted by them back to the attacked website.

2) Security Question: Another identification challenge is presented by security questions. During the registration, users are sometimes asked to answer personal question(s) that will be used to identify them in case the password is lost or forgotten. When the attacker receives a security question in the password reset process, he can just forward this question to the victim who is currently registering to the attacker's website. The attacker will forward the user's answer on to the attacked website.

3) Code to the Mobile Phone: Authentication can be done via one of three approaches: (1) something you know (e.g., password), (2) something you are (e.g., fingerprints), and (3) something you have (e.g., special token device or a phone). Therefore, when users forget their password, many websites allow them to authenticate themselves via something they have, like a mobile phone. This is usually done by sending a message with a password reset code to the phone of the user via SMS. Some websites also support an automated phone call to the user, in which the code is given. The user is required to insert this code in order to change her password. In Section IV, we analyze the different messages sent by popular websites and show that it is possible launch a PRMitM attack also in this case. In Section V, we show that phone calls are also vulnerable to the attack.

Name	Global rank	Email link	Phone code	Security question	CAPTCHA
Google	1	√	√	√	
Facebook	2	√	√		
Youtube	3	Uses Google account			
Baidu	4	√	√		√
Yahoo	5	√	√		
Wikipedia	6	√			√
Amazon	7	√			
QQ	8	√			
Twitter	9	√	√		
Live & Bing & Outlook	10	√	√		√
Linkedin	18	√	√		
Ebay	25	√	√		
Netflix	37	√	√		
Paypal	41	√	√		

TABLE II: Challenges used in password reset process by the 10 most popular sites [27] and other popular websites.

4) Reset Link to the Email: The most common countermeasure involves sending a link to reset the password of the victim's email address. To bypass this mechanism, the attacker must be able to access data in the email account of the victim; therefore, the PRMitM attack cannot be applied on websites that allow password reset only by sending a reset link to the email. Unfortunately, this option is usually not relevant for the email services themselves. Moreover, relying only on this option blocks password recovery when users have lost access to their email account.

C. Challenges in Popular Websites

[1] We surveyed the challenges used during the password reset process by the most popular websites in the world [27]. Table II summarizes the findings. The 10 most popular websites support password reset using the user's email account and most of them allow password reset using a phone as an alternative.

[1] The challenges survey that is summarized in Tables II and III was conducted during the second quarter of 2016.

Name	Global rank	Email link	Phone code	Security question	CAPTCHA
yandex.ru	20	√	√	√*	√
mail.ru	27	√	√		√
aol.com	152	√	√		√
gmx.net	232	√	√		
rediff.com	334	√	√	√	√
iCloud.com	353	√			
zoho.com	589	√			
mail.com	1505	√		√	√
gmx.com	5204	√		√	√
fastmail.com	6305	√			

TABLE III: Challenges used in popular email services that do not appear in Table II. (*) Yandex supports password reset using security question only for users who did not set phone number and alternative email address.

Google is the only one that also supports security questions, and three of them require solving a CAPTCHA in addition to one of the first two challenges.

We also surveyed popular email-services, because those have difficulty offering an email-based password recovery process. Email-services are usually very sensitive; by obtaining access to the victim's email account, an attacker can further reset the password of other websites.

The challenges used by popular email-services that do not appear in Table II, are summarized in Table III. We chose only email services to which we could register, all of them from USA, Russia, India, and Germany.

Among these 10 email services, we found that Yandex, one of the most popular websites in the world, mail.com, gmx.com and reddif.com allow password recovery by only answering a security question and solving a CAPTCHA. In Yandex, this option is possible only for users who did not input their phone and alternative email. This makes these websites vulnerable to a simple variant of the PRMitM attack, in which the attacker only forwards the security question and the CAPTCHA challenge to the victim to solve, and then takes over the account.

Google also supports password recovery using security questions. However, Google's mechanism is mainly based on activities done by the user in the account, and on other parameters like the IP address and the browser used by the requester. Although Google also uses general security questions in some cases, PRMitM attack alone cannot be used to overcome the security questions. See also Section VII-A.

Clearly, most of the popular websites and email services support authentication using a mobile phone. In Sections IV and V, we show that sending the reset password code by SMS or phone call is also vulnerable to attack.

D. Evaluation: PRMitM with Security Question

As some websites still allow password reset that relies on security questions, we conducted a small user study (Experiment 1) to test whether or not users provide the correct answers for such questions. Since popular websites do not rely on security questions, we could not recruit participants and simulate a real attack on their accounts.

Yet, under the assumption that users who give the correct answer in a low-importance website would also correctly answer their security question in more reputable websites, the experiment should offer a good indication. Although not analyzed in this experiment, users who give the same wrong answer to both the attacked and the attacking websites, are vulnerable to the attack.

EXPERIMENT *1:* Correctness of security question's answer.

Experiment process. Participants were asked to register to a website in order to perform a short experiment. During the registration process, they were asked to type their email address, and only then, to answer a classical security question: What is your mother's maiden name. Once the users completed the registration, we asked them whether the answer they just typed was correct.

Ethics. We did not save any private data about the participants. We only saved the answer distribution of the last question.

Participants. 52 volunteer students from our institute.

Results. Although registering to a low-importance website, 76.9% of the participants provided the correct answer to the security question.

Bonneaue et al. [28] conducted a larger survey with the participation of 1500 users. There, 37% of the participants reported that they gave wrong answer to the security question when registering on their primary email account. Beyond the population and the number of participants, the difference in the results can be due to the experiment process.

In our experiment, the users *answered* a security question; in [28], the users were only asked about registration that probably occurred several years ago. It is surprising that the survey of [28], did not include statistics about users that do not remember their answers. For example, the authors of this paper do not even remember if they were asked to answer a security question during their registration to Gmail.

Even if only 63% of the population are vulnerable to the attack [28], this is still a high percentage and an indicator for the problem of relying on security questions.

IV. PRMITM VIA SMS

Popular websites also usually offer mechanisms for password recovery to users who lost access to their email account. The problems with security questions [29]–[32] and the popularity of mobile phones has made the authentication using mobile devices a preferred option for password recovery (e.g., see Tables II and III). The most common way to authenticate a user via mobile phone is by sending a code to the device. The user then has to insert the received code into the website to reset the password.

Unfortunately, in some cases, when the reset code is sent by SMS, the PRMitM attack is still possible. The attacker asks the victim for her phone number, claiming that a code will be sent to it. Then the attacker initiates a password reset process using this phone number in the attacked website, causing this website to send an SMS with a password reset code to the victim's phone. The victim receives the expected message, and may type the code in the attacking page. Now, the attacker can complete the password reset process.

The attacker can even trick the user into disclosing her password reset code under simpler conditions. Unlike security questions, a code to the mobile phone is not used solely for registration and password recovery. Although email addresses that can be generated easily and for free by bots, mobile numbers are harder and more expensive to attain. Therefore, sending a code to a mobile device is a reasonable way to both prove that users are not bots and to prevent overuse by users. Instead of the registration process, the attacker can ask the user to insert a code sent to her mobile phone before accessing a resource or downloading a file.

In the rest of this section we discuss the problems with password reset using SMS (Section IV-A), survey this mechanism in popular websites (Section IV-B), and ultimately evaluate the attack on Facebook users (Section IV-C).

A. Limitations of Password Reset Using SMS

We identified several problems with sending a password reset via SMS. While the first problem is inherent, we found additional problems that appear in some of the websites and can be easily fixed.

Unclear message. SMS is limited to 160 ASCII characters, and there are at least 3 pieces of information that should appear in each message in addition to the password reset code: (1) the sending website, (2) explanation about the code's meaning (password reset), and (3) a warning to avoid disclosing the code to anyone else. Most of the websites are aware of the need to include these three elements. As evidence, they include all of them (and more) in emails that are sent to reset a password. Yet, the length limitation and the desire to avoid sending multiple SMS messages prevent them from sending the optimal message.

Sender identity. SMS spoofing is the process of setting the sender of SMS messages to a value that is not the originating mobile number. The sender can be set to another number or to alphanumeric text. Usually, SMS messages are sent from numbers that are not known to the users. Using SMS spoofing, the sending companies can give the user an indication about the sender. However, we noticed that some of them do not use this option at all, or they use it with a sender name that is non-informative. In spite of that, the importance of using informative sender identity seems to be minor compared to content of the message; see the results analysis of Experiment 2.

Token validity period. When a code is given, the user can use it only during a limited time period. However, this time period varies between websites, and can be anywhere from 15 minutes to 24 hours. In the PRMitM attack, this time slot is critical. Ideally, the attacker would like to reset the passwords as late as possible. An attacker who gets the code at noon would prefer to reset the password late at night, when the user is sleeping.

Language compatibility. Many websites offer services in many languages, but some do not send the SMS message in the supported language. Users who cannot read and understand the text, but only to identify the code, become exposed to the attack. Namely, users who get a message in an unfamiliar language, can read the code, but not the attached text. In such cases, an informative warning text becomes irrelevant.

B. Websites Survey

Table IV summarizes the SMS messages sent by popular websites during their password reset process. We also specify which text represented the sender, the code's validity period, and whether the language is adjusted to the user.

The table presents only websites that support multiple languages. The second column shows the English message sent in the SMS by each of the websites.

Unlike common password reset emails, *none* of the websites' SMS messages included a warning about the danger of disclosing the code. The fact that this message was sent as part of a password reset process appears in only 4 of them. Popular websites like Yahoo and Google have a general message about verification codes. Such a message can be easily abused by a PRMitM attacker. Moreover, unlike their messages in the other languages, both Google and Yahoo send non-secure SMS messages to Russian language users. Their Russian message simply says *"Your verification code: XXXX"*, without any indication to the sender in the message body.

Another vulnerable website is Yandex, the only website we tested for which none of the SMS messages contain the name of the website. Yandex simply sends a verification code and asks the user to enter it in a text field.

To detect what appears as the SMS sender, we initiated password reset process using SMS from three different devices. Only three websites noted the name of the website as the sender. In the SMS from Facebook, the sender appeared either as a number or as Facebook. In all the other cases, we received the SMS from an unknown number or got the string "Verify" as the sender.

To test the validity period of the received code, we initiated the password reset process and tried to use the code after different time periods. We could not find the exact expiration time, but tried different values and noted the longest time period after which we succeeded in using the code. For services that do not specify the expiration of their code, we tested the following time periods following a binary-search based approach: 5, 10, 15, 30, 45, 60 and 90 minutes, and $2, 3, 4, 6, 8, 10, 12, 18$ and 24 hours.

To test language compatibility, we tested the accounts against several popular languages they support. Specifically, we tested: English and Spanish, which are very common languages; Russian and German, which are common; and Hebrew, which is not a common language.

We say that a website is *SMS language compatible (SLC)* with a language if it sends the password reset message in

Site	SMS text	Sender	Validity period	Language compatibility
Google & Youtube	Your Google verification code is XXXXXX	Google	90 minutes	Full
Facebook	XXXXXX is your Facebook Password reset code or reset your password here:https://fb.com/l/YYYYYYYYY	Facebook/Number	10 hours	English only
Yahoo	Your Yahoo verification code is XXXXXX	Verify	15 minutes	Full
Twitter	Enter this code to reset your Twitter password: XXXXXX	Number	60 minutes	English only
Live & Bing & Outlook	Use XXXXXXX as Microsoft account password reset code	Verify/Number	15 minutes	Full
Linkedin	Your LinkedIn verification code is XXXXXX.	Verify/Number	15 minutes	Good
Yandex	Your confirmation code is XXXXXX. Please enter it in the text field.	Yandex	2 hours	Full
Ebay	Your single-use eBay PIN is XXXX	Number	24 hours	Partial
Mail.ru	MailRu: XXXXXX - password recovery code for usern***@mail.ru	MailRu	45 minutes	Full
Netflix	Your Netflix verification code is XXXXXX	Number	15 minutes	Full

TABLE IV: Password reset by SMS in popular websites.

this language. We tested whether a website is SLC only with regards to *supported languages*, which are languages in which the website gives services. We gave one of four grades to websites for their SMS language compatibility.

1) *Full*. The website is SLC with all of its supported languages that we tested.
2) *Good*. The website is SLC with all of its supported *common* languages that we tested, but not SLC with an uncommon supported language.
3) *Partial*. The website is SLC with more than one supported common language that we tested, but is not SLC with another supported common language.
4) *English only*. Although supporting also other common languages, the website is SLC only with English.

Six out of the 10 websites in Table IV were assigned a *Full* grade. This means that some users of the other four may receive an SMS they cannot understand, which makes them an easy target for PRMitM attacks. We tested the websites by configuring the accounts to use each of the languages. Because some websites may determine the language according to parameters such as the country prefix of the phone number, a non-*Full* grade does not mean the website does not send SMS in some of the languages. However, by itself, sending critical messages in a language that is different from the language the user chose is a problem.

C. Evaluation

In the survey we conducted (Section IV-B), we found three types of messages; none of them explicitly warn the users against typing the code in another website. The messages are sorted from the most to the least vulnerable.

1) **Just a code**. Message contains only the code, without mentioning both the reset process and the sending website. For example: Yandex, Google and Yahoo in Russian.
2) **Sender and a code**. The sending website is mentioned with the code, but there is no evidence of the password reset process. For example: Google, Yahoo, and LinkedIn.
3) **Password reset code message**. In addition to the code, the password reset and the sending website are mentioned. For example: Facebook, Twitter, and Microsoft services.

In a typical PRMitM attack that abuses the password reset using SMS, the attacker asks the users to authenticate themselves by sending them an SMS. Once the attacker gets the phone number of the victims, he initiates the password reset process for their phone numbers in the attacked website. If the victims receive the code and type it into the attacking page, the attacker can take over their accounts in the attacked website.

Naturally, SMS messages of the third type are harder to abuse for the PRMitM attack. Experiment 2 shows that it is still possible to effectively abuse such messages, and that a more detailed SMS message does not provide full protection.

Due to ethical reasons, we did not use the SMS code to complete the password reset process on the accounts of the participants. To make sure the SMS code is enough for the attack to work, we successfully simulated the attack under experimental conditions on several of our own accounts. We showed that it is possible to initiate the password-reset process from a machine that has never been used before for the attacked account as tested in the experiment, and that it is possible to complete the attack with the code (that the victim gets to his phone and forwards to the attacker). Furthermore, in the examined case of Facebook, it is also possible to use the code to gain access to the account, without resetting the password. In this case, no notification about password-reset is sent to the email of the victim.

It is important to note, that in the experiment, the attacking machine was located in the same country as the attacked computers. In practice, the attacker can detect the IP address of the victim and launch the attack from a machine under similar settings.

EXPERIMENT *2:* Effectiveness of PRMitM attack on Facebook users using SMS and comparison between Facebook's SMS and more detailed SMS.

Experiment process. Participants were invited to an experiment about memory skills. Before they accessed the experiment webpage, they were told that if they encounter any problem or something they do not like, they are free to stop the experiment, go directly to the final form, and leave feedback about the experiment process. The experiment page that was actually the attacking page asked them to identify themselves using their phone number. Specifically, the page asked the participants to type their phone number, so they can receive an SMS with a code that should be typed in. Each user was randomly assigned either to the *Facebook SMS* group or to the *detailed SMS* group.

In the *Facebook SMS* group, once the user typed her phone number, the attacking page contacted a server that sent a request to Facebook for password reset via SMS. Facebook then sent the message to the participant. Our server was implemented in Python and used Selenium to imitate browsing activity to Facebook's servers. In the *detailed SMS* group, we spoofed the following SMS from Facebook: **WARNING* Someone requested to reset your Facebook password. DO NOT SHARE THIS CODE with anyone or type it outside Facebook. The password reset code is XXXXXX*.

If the participant identified the threat, she could stop the experiment and move to the final form. Other participants simply played a memory game for 90 seconds before they were redirected to the final form.

In the experiment's final form, we gradually asked the participants about their feelings and suspicions. The users were told that the experiment's participants were randomly divided into two groups, and that half of the participants were manipulated. We then asked them which group they thought they were assigned to. In reality, all the participants were manipulated according to their group.

After that question, we continued hinting to the participants about the real purpose of the experiment, by telling them that the goal of the discussed manipulation was to take over one of their accounts. We then asked again which group the participants thought they were assigned to.

Before asking this question the third time, we told the users that the account we tried to hack was a Facebook account.

Ethics. We had a dilemma about the right way to conduct this experiment. We could spoof the Facebook messages and avoid contacting Facebook for the *Facebook SMS* group. However, we chose to simulate a real attack, mainly because the interaction between the attacking page and the attacker's server, and between the server and Facebook, takes time and could arouse suspicion. We wanted to make sure the experiment simulates a real PRMitM attack, and prove that this attack is indeed practical in real world conditions. We did not take over any Facebook accounts, nor did we save the codes typed by the users. We only verified the correctness of the typed codes with the users.

Participants. From our institute, 88 volunteer students participated in the experiment. Of them, 42 were assigned to the *Facebook SMS* group and the others to the *detailed SMS group*. We used volunteers on purpose, so they could feel free to leave the experiment at every moment. The participants did not take a part in other experiments or surveys conducted in this research.

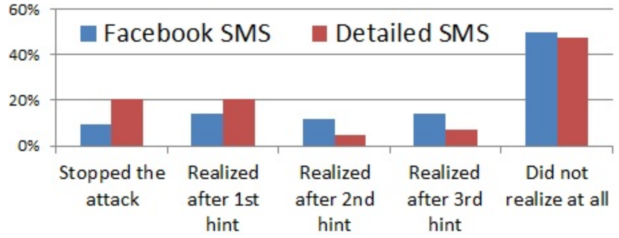

Fig. 2: Results of Experiment 2. Only 9.5% of participants in the *Facebook SMS* and 20.5% in the *detailed SMS* group detected and stopped the attack. In both the groups, about half the participants did not realize they were attacked; the others realized after some hints.

Results. We completed the attack successfully on 90.5% of the *Facebook SMS* group, and on 79.5% of the *detailed SMS* group. Namely, among the users who underwent a simulation of the attack, only 4 participants stopped the experiment and avoided sending their Facebook password reset code to our server. In both groups, around 50% of the participants did not realize they were attacked even after we told them we hacked into the Facebook account of half the participants. We observed that the hints helped the participants understand what happened, but those in the *detailed SMS* group were quicker to suspect a security issue. Figure 2 depicts the results.

Results analysis. The results show that the PRMitM attack can be launched automatically.

We questioned participants who did not stop the attack in order to understand their behavior. We gained two important insights that are relevant for improving the password reset process:

1) Many users just searched for the code without reading the text. Some of them did not open the message, but read the code from the notification that was prompted in their phone.
2) Many users who noticed that the message was sent from Facebook, thought the login to experiment was done using the widely used *login with Facebook* mechanism [33]. This means that the sender identity as specify by SMS spoofing has a minor importance in the attack, mainly if the content of the message is unclear. Furthermore, adding sentences to the attacking page like ”Powered by Facebook” or even just an explanation that the message will arrive with specific sender, may make SMS spoofing even more worthless.

Relying on this feedback, we designed mechanisms that will prevent such phenomena. See Section VII-B.

V. PRMitM via Phone Call

This section discusses PRMitM attacks that exploit password reset using phone calls. We first compare the use of SMS and phone calls in password reset processes, and then describe the vulnerabilities we found. Finally, we bring Google, the most popular website in the world, as an example to vulnerable website and evaluate the PRMitM attack on Google users.

A. SMS code vs. Phone Call

There are many comparison parameters between password reset process using SMS and phone call. This section focuses on security aspects, mainly considering the PRMitM attack.

Sender identifier. Using SMS spoofing it is possible to give an indication about the sender regardless of the content. In phone calls, there is no such equivalent mechanism and the phone calls arrive from unrecognized numbers.

Length of message. SMS code is limited in its length, and hence usually does not contain enough information (see Section IV-A). In phone calls it is possible to deliver longer messages.

User attention. Reading a code from SMS does not require effort or concentration. Actually, in Experiment 2, we noticed that some users do not open the message, but read the code from the notifications bar. Other users read only the code. In a phone call, the user dedicates more attention to the content of the phone number, mainly because the user will not have access to the code once the phone call ends.

Language issues. Reading a reset code from an SMS in unknown language is possible, as numbers are written the same in many languages. Even a code that combines letters can be differentiated from the other letters in the message. Therefore, in many cases, companies send SMS messages in a language that is different from the language that the user uses. Such cases can be exploited by the PRMitM attacker. To extract the reset code from a phone call, at least basic understanding in the language is required; hence, a user that extracts the code from a phone call is more likely to also understand the message.

Interactivity. Interactivity in the password reset process can be used to ensure that the user understands the situation. Phone calls are more suitable for such an interaction, e.g., by typing digits; indeed, Ebay uses interactive phone call to deliver the password reset code. It is much harder to create secure interaction using SMS.

B. Vulnerable websites

Websites that support password reset using phone calls might be vulnerable to PRMitM attack similarly to the SMS variant. Like SMS messages, a secure phone call must include the initiating website, the reset password process, and a warning about disclosing the code.

If a website uses a phone call that just reads the reset code, the PRMitM attacker can ask for the phone number for calling the victim, and instead of that to initiate a password reset process against the website using the victim's phone number. The website will call to the victim, but without any option for the victim to detect the source of that call. Hence, without suspecting, the victim will forward the received reset code of the attacked website to the attacker.

Among the popular websites surveyed in this paper (Top 100 websites [27] that appear in Tables II and III), only Google, Linkedin, eBay and Netflix support password reset using both SMS and phone call (in our country). Paypal supports only phone calls.

Among these 5 websites, we found that Linkedin and Google are vulnerable. Linkedin's phone call does not mention Linkedin at all. In Google, we noticed a difference between the 10 languages we could test.

The phone calls in German, French, Russian, Italian, and Persian are just a translation of the English call (hence will be denoted as the *English* group):

> *Hello! Thank you for using* **Google** *phone verification. Remember! You should not share this code with anyone else, and no one from* **Google** *will ever ask for this code. Your code is XXXXXX. Again, your code is XXXXXX. Good bye.*

However, the set of *vulnerable* phone calls in Spanish (second most popular language in the world, more than English), Arabic, Dutch and Hebrew are surprisingly vulnerable:

> *Hello! Thank you for using* **our** *phone verification. Your code is XXXXXX. Again, your code is XXXXXX. Good bye.*

The phone calls in the *English* group mention the sender (Google) twice. They also contain a warning about sharing the code. However, they do not explain what is the meaning of the code; namely, the password reset process is not mentioned.

In the *vulnerable* calls, the sender identity is replaced by the general word *our*, and the warning is omitted. Because there is no indication to the real sender or the real meaning of the received code, the phone calls in these languages are completely vulnerable to PRMitM attacks.

C. Evaluation: PRMitM on Google Phone Calls

This section describes Experiment 3, a user study we conducted to evaluate the PRMitM attack on Google users, exploiting the password reset process via a phone call.

Due to ethical reasons, we did not use the codes received in the phone call to complete the password reset process. However, similar to Experiment 2, we successfully tested the possibility to complete the password-reset process on several of our own accounts. Namely, we verified that under the experiment conditions, in which a password reset request is sent from a machine that was not used before for the attacked account, it is also possible to successfully reset the password.

Experiment *3:* Effectiveness of PRMitM attack on Google users using phone calls.

Experiment process. The experiment process was the same as in Experiment 2. However, instead of telling the users that they will get a code in SMS, we told them that we will call them. To initiate a password reset process in Google, only the email address of the victim is required. However, we asked the

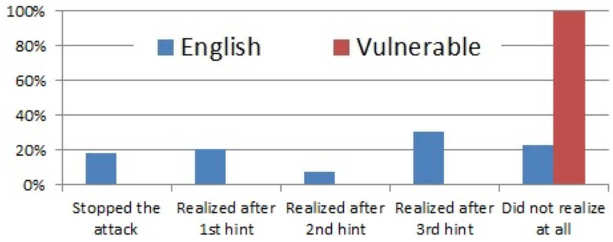

Fig. 3: Results of Experiment 3.

users to insert both their email address and phone number, so the call will not be suspicious. Once the users inserted their phone number, our server contacted Google and initiated a phone call to them in order to reset the password. We did not know in advance which language is used by the users, but asked for this information in the final experiment form.

Ethics. The dilemma from Experiment 2 remained with us also in this experiment. From similar reasons, and because we could not predict the call language of some of our participants, we decided to initiate a phone call from Google and not a spoofed one. As done in Experiment 2, we did not save the codes typed by the users, and only verified their correctness with the users.

Participants. 68 volunteer students from our institute, 39 from them used the English language (*English* group), and the others used languages that have vulnerable phone calls (*vulnerable* group).

Results. As expected, due to lack of any indication about the real source of the call, all the participants of the *vulnerable* group completely failed to detect the attack. Among the participants of the *English* group, only 7 participants (18%) blocked the attack. 59% were ”attacked” successfully and realized after one of the hints. The other 23% did not realize that they were manipulated also after the three hints. Figure 3 depicts the results.

Results analysis. While we expected that for the languages used in the *vulnerable* group it will be impossible to detect the attack, we were surprised how vulnerable is the English phone call. Although the number of participants was low, the results clearly indicate that even the English phone call is vulnerable to the PRMitM attack.

We were mainly interested in users from the *English* group who failed to stop the attack. The most common argument was the fact that the phone call did not specify anything about the meaning of the code. To the users who thought that the code comes from other websites, it sounded reasonable that no one from Google will ever ask for this code. A few users said that they did not give enough attention to the message code. Relying on this feedback, we designed and evaluated a secure phone call that will prevent the attack; see Section VII-C.

VI. Mobile Applications Vulnerabilities

The previous sections presented several variants of the PRMitM attack. All the attacks were demonstrated and evaluated on popular websites.

Although websites are easy targets, it is possible to attack other applications as well. In particular, some mobile applications require authentication that is done by typing a code that is received via SMS or a phone call. This makes them potentially vulnerable to the PRMitM attack, if the content of message is not clear enough.

We audited some of the most popular messaging applications available today to get indication about possible vulnerabilities. This section brings our short survey and summarizes its findings.

A. Survey: Password Reset in Mobile Messaging Applications

The vulnerabilities we found in popular websites encouraged us to search for similar vulnerabilities also in mobile applications.

In particular, we chose to audit the password reset process of messaging mobile applications. Taking over such applications exposes private and sensitive information about the user, and allows the attacker to perform sensitive operations like sending messages in the name of the user.

Table V lists the applications we tested and the supported channels for password reset process [2].

Mobile applications are especially interesting in the perspective of password reset process, as messages with password reset code can be sent through the applications themselves to the mobile phone of the user. This is an additional option to initiate password reset process that does not suffer from the limitations of SMS and phone calls (e.g., limited length, graphic, etc.). Namely, an installed mobile application can easily explain to the user about the password reset process; see also Section VII-E.

Among the nine very popular applications we tested, only Telegram supports password reset via the application. Telegram also tries to use this option to reset the password before other techniques like SMS or phone call are used.

SMS is the most supported way to initiate password reset process. Only four applications support password reset only via Email, three of them exclusively, which makes the PRMitM attack impractical on them.

B. Mobile Applications PRMitM Vulnerabilities

In addition to the lack of use in the application itself to reset the password, we found the following vulnerabilities:

Vulnerable phone calls in Whatsapp, Snapchat and Telegram. Among the applications we tested, all the three that use phone-call during their password reset process, are vulnerable. Namely, in the phone calls of Whatsapp, Snapchat and Telegram, there is neither indication to the source of the call nor explanation about the meaning of the received code nor warning about not giving away the code. See Table VI.

In Snapchat, to initiate the password reset code, the attacker has to solve a CAPTCHA and to get the username. While using the PRMitM attack to solve the CAPTCHA seems reasonable, it seems harder to trick the victim to give his

[2] The survey was conducted during the third quarter of 2016.

Application	Email	SMS	Phone call	Application message	Remarks
Whatsapp		√	√		Phone call can be initiated only 5 minutes after the SMS was sent
Facebook Messenger	√	√			Password reset is done as in Facebook accounts
Telegram		√	√	√	A message is sent through the Telegram application. If no code has been sent by the user, an SMS is sent. If yet, no code has been sent by the user, a phone call is done.
Kakao		√			
Kik	√				
Line	√				
Nimbuzz	√	√			The user is required to solve a CAPTCHA and insert the username
Skype	√				
Snapchat	√	√	√		The user is required to solve a CAPTCHA like game

TABLE V: Options to reset passwords in popular messaging applications

Application	Phone call message
Snapchat	Your confirmation code is XXXXXX. again: XXXXXX
Whatsapp	Your verification code is XXXXXX (repeated four times)
Telegram	Hello your code is : XXXXXX. once again: XXXXXX

TABLE VI: Messages used in phone calls during password reset process of popular messaging applications

Application	SMS message
Whatsapp	Your WhatsApp code is XXXXXX but you can simply tap on this link to verify your device:v.whatsapp.com/XXXXXX
Facebook Messenger	XXXXXX is your Facebook Password reset code or reset your password here:https://fb.com/l/YYYYYYYYY
Telegram	Telegram code XXXXXX
Kakao	XXXXXX Verification Code from KakaoTalk. [KakaoTalk]
Nimbuzz	Your Nimbuzz account password is : XXXXXX
Snapchat	snapchatcode: XXXXXX.happy snapping

TABLE VII: SMS messages used in the password reset process of popular messaging applications

Snapchat username. Yet, the attacker can launch targeted attacks on users whose username is known to the attacker (e.g., by applying social engineering techniques [13], [20], [34]).

In Whatsapp, the attacker cannot initiate the phone call immediately. Whatsapp's password reset process begins with an SMS that is sent to phone number that is used in the process. The phone call is initiated only 5 minutes later, if the process has not completed. Although the SMS used by Whatsapp is also vulnerable (see below), this limits the effectiveness of the attack. E.g., for attackers that can block SMS messages, or only for users that will not correlate the SMS from Whatsapp with the registration to the attacking page that claims to call him, and to the vulnerable phone call that will be received later (the attacking page can mention that it usually takes 5 minutes until the call is received).

Telegram's password reset process is similar to that of Whatsapp. However, the phone call is initiated only if the user does not respond to a message that is sent to him via the Telegram application or later via SMS.

Non-informative SMS in all of the applications. The SMS messages of all the applications contain the name of the application. Yet, none of them contain a warning that will prevent the user from typing the code in other website. Following the results of Experiment 2, this puts their users in risk.

This becomes more critical due to lack of language compatibility. The surveyed applications are widely used across the globe, with many users who use different languages. In spite of that, except Whatsapp, the messages were sent only in English, regardless of the language settings or the language used by users. The lack of language compatibility increases the chance that users will just check for the code without reading the other content of the message. This problem is relevant to Facebook Messenger, Telegram, Kakao, Nimbuzz and Snapchat.

The SMS messages used by the surveyed applications (Table V) appear in Table VII.

VII. Defenses

This section discusses defenses against the PRMitM attacks introduced in the previous sections. There are multiple ways to defend against each of the attacks; some of them can be implemented in several ways. The evaluation of all the defense techniques and their different variants deserves a separate work. The variants of each countermeasure should be evaluated in user studies to learn about the optimal configuration for each of them.

The main scope of this paper is to introduce the attack, and to provide first aid that can block it. Therefore, we mainly discuss and evaluate two countermeasures, which we believe can be easily deployed by websites. Both the techniques force the users to understand that someone asked to reset the password. Because more efforts are required, it might be claimed that these mechanisms harm the user experience. However, we believe that in operations like password reset, it is completely reasonable to make the users work hard to reset their password if it significantly improves the security.

A. Good Security Questions

Security questions that are not exclusively related to the website might be vulnerable to PRMitM attacks.

If a website asks many questions that are directly related to the actions done by the user in that site, they cannot be forwarded to the user as legitimate security questions for other websites.

Google is an example of a website that relies on security questions combined with other parameters such as IP addresses and originating browser. In addition to general security questions, Google asks questions about common contacts, user-defined labels, and the use of multiple Google services.

Nevertheless, it is desirable to avoid relying on security questions, as they can be bypassed by attackers, especially if the attacker is related to the victim.

B. Secure Password Reset Using SMS

Section IV showed that some users do not read the entire SMS messages they receive (Experiment 2). Beyond that, current SMS messages (Table IV) lack a warning about giving away the code, and are sometimes missing explanations about the meaning of the code and the sender. Lack of language compatibility makes this problem even more serious.

Following our findings, we believe that a password reset code should not be sent in a clear text over SMS. Hence, we designed a link-via-SMS (LVS) password reset procedure, and evaluate it compared to detailed SMS messages.

1) Link-Via-SMS (LVS) Password Reset: Links for password reset are used mainly when the password reset is done via email accounts. Among the websites we surveyed, only Facebook sends a link to reset the password in addition to the code.

Sending a detailed SMS message with a long link (instead of a code) overcomes the limitations of the SMS with the code. First of all, to exploit such a message, the PRMitM attacker has to ask the user to copy a link to his website, which is unusual. Moreover, since the link is long, the attacker cannot just glimpse at the message. This increases the likelihood that the victim will notice the rest of the text.

A long link is better than just a long code. The natural user interaction with links is to press on them. On the other hand, there is always a chance that a user will just copy the code without reading the message.

In our implementation of the LVS, the link refers the user to an interactive page that has an alert about the attempt to reset the user password.

The user experience might be degraded if the user cannot access the Internet from her phone. However, we believe that in such cases, it is reasonable to force the user into typing the long link into her browser's address bar.

Another question that should be discussed is whether LVS increases the risk to other attacks. We believe that the answer to this question is negative. Following received links in SMS might be harmful [35], [36], but this has nothing to do with an SMS that is sent by a service that intends to protect its users. Attackers might try to impersonate legitimate LVS message to trick users to follow malicious links; however, they can do the same also for legit SMS messages (although the original message do not include a link).

2) LVS Evaluation: Experiment 4 repeats Experiment 2 but with an LVS instead of the classical SMS with the code.

EXPERIMENT *4:* Effectiveness of LVS against PRMitM attack on Facebook users.

Experiment process. The experiment process was similar to Experiment 2 with a single change: We sent the participants an SMS with an LVS message.

The LVS message was: **WARNING* Someone requested to reset your Facebook password. Press this link to reset your Facebook password: http://bit.ly/XXXXXXX. DO NOT SHARE IT!*

Ethics. We only verified that the users indeed have a phone number related to their account. We did not contact Facebook to initiate a password reset process for the participants' accounts.

Participants. 46 volunteer students from our institute that did not participate in any other experiment or survey.

Results and analysis. All the participants stopped the attack; namely, none of them typed the link into the attacking page. This reinforced our hypothesis, that LVS is indeed a secure way to reset a password using SMS. This is important due to the poor results achieved by the classical SMS messages (see Experiment 2).

C. Secure Password Reset Using Phone Call

Although phone calls were shown to be vulnerable in Experiment 3, they can be used effectively and securely for password reset processes. Two elements must hold: (1) the message must include the sender, the meaning of the code, and a warning about misuse, and (2) the call must cause the user to listen and understand the message. For this purpose we conducted Experiment 5, which is similar to Experiment 3, but evaluates more detailed and interactive phone call. The results show that indeed, such a phone call significantly improves the results.

EXPERIMENT *5:* Effectiveness of detailed and interactive phone call against PRMitM attacks.

Experiment process. The experiment process was the same as Experiment 3. However, instead of initiating a phone call from Google, we called the users with an (interactive) phone call. We denote by X_i and Y_i randomly chosen numbers such that $X_i \neq Y_i$. Pressing Y_i always leads to *Good bye! Consider securing your account!*. X_i leads to the next sentence.

1) Hello! This is a phone call from Google in order to reset the password of your Google account. Click X_0 if you expected this call, and Y_0 otherwise.
2) Warning! Someone asked to reset your Google password. I repeat: Someone asked to reset your Google password. If you did not ask for a password reset code, press Y_1; otherwise, press X_1.

3) You are about to get a code to reset your Google account password. You should never share this code with anyone else and never type it in other websites. No one from **Google** or other legitimate websites will ever ask for this code. Your code is XXXXXX. Again, your code is XXXXXX. Good bye.

In each of the choices either X_i or Y_i will be read first randomly. For example, in step 1 of some of the calls, instead of mentioning X_0 and then Y_0, the following sentence was read: *Click* Y_0 *if you did not expect this call, and* X_0 *otherwise*. Without waiting more than a second for a user to press something, our phone call lasts about 70 seconds, double that of Google's current English phone call.

Ethics. We did not initiate the password reset process for the participants' Google accounts and did not save their details.

Participants. 45 volunteer students from our institute that did not participate in any other experiment.

Results and analysis. None of the participants disclosed their code, which shows that such a phone call is very effective. Some users failed to follow the instructions the first time. Namely, they initiated the phone call two or three times until they realized that they should not use this phone call to get a code for the experiment website. Although it might occur also for users who really want to reset their password, we believe that the users will agree to bear this overhead to enhance their security.

D. Notifications

Websites should notify their users about both password reset requests and upon password change. The notification should be done both by sending an email and by sending an SMS. This is especially critical when the password reset is done using the phone, and even more crucial for email services. If the attacker takes over an email account, he can delete the received notification. Similar to the password reset messages, the notifications must be clear.

Among the websites we tested (Tables II and III) that support password reset using a phone, only Google sends an SMS notification after a password change.

E. Alternative Countermeasures

A secure password reset process can be implemented using a phone via either SMS or phone call. An additional phone method implemented by Google relies on applications installed on the user's phone. An application can prompt a clear notification and initiate a password reset process that does not involve any external website. This makes the process immune to PRMitM attacks.

Another alternative for users who do not have an account is to rely on the accounts of friends [37]. The user should give in advance email addresses or phone numbers of x friends. In the password reset process, each of the friends will get a code. $y \leq x$ of the codes are required to reset the password.

VIII. Related Work

In this section we describe both MitM attacks in the application layer, and other techniques that can be used to overcome some of the challenges in the password reset process.

A. Application Level MitM

In the attacks described in this paper, the attacker manipulates the victim into solving challenges raised to the attacker by websites. Previous work offered similar approach to solve CAPTCHA challenges. Egele et al. [38] offered to overcome CAPTCHA challenges prompted by websites by prompting the same CAPTCHA challenges to visitors of other websites under the attacker's control. Similarly, viruses and botnets like Koobface enforced the users of infected computers to solve CAPTCHA challenges for them [39].

Lauinger et al. offered to perform MitM attack between two chatting clients, by opening a chat with each of them, and forwarding their input text from one chat to the other [40]. That way, the attacker can automatically launch social engineering attacks without designing advanced artificial intelligence bots [41].

Another form of MitM attacks is man in the browser (MitB) attacks [42]. In MitB attacks, malware takes over the browser and acts as a proxy between the user and the web. That way, the malware can obtain every piece of information typed by the user. Moreover, the attacker can manipulate operations done by the user. For example, to change the recipient of financial transactions.

Another approach to gain a MitM capability that includes manipulation on the user, is to lure the victim to use a router controlled by the attacker. The most known attack is the *evil twin* attack [43], [44]. In the evil twin attack, the attacker creates a WiFi access point with an innocuous name, possibly a name of a trusted WiFi access point. The attacker eavesdrops HTTP connections of victims who connect to his rogue access point and to launch phishing attacks on them.

Phishing attacks also load content from the websites to which they impersonate, creating kind of MitM between the original websites and the clients to be as similar as possible to the original websites [21]. More than a decade ago, sophisticated phishing attack was used to bypass anti-phishing system used by Bank of America [22]. In the attack, a login phishing website acts as a MitM between the user and the login page of the financial institution, forwarding the challenges to the user and their solutions to the bank. However, this is still a phishing attack and it is not different from other phishing attacks that impersonate a login page and imitate the login procedure. The PRMitM attack shows that such techniques are possible even without the need of impersonation to other websites, which is the greatest challenge in phishing attacks. See more on the difference between phishing and PRMitM attacks in Section II-B.

Finally, in Section VII-B, we argue that during password reset process, links should be used instead of codes. The authors of [45] recommended to use links in registration process due to similar reasons.

B. Overcoming Password Recovery Challenges

During the password recovery process websites use several challenges. Some of these challenges were analyzed in previous work.

Although a human attacker can solve CAPTCHA challenges or use cheap labor [46], it is desirable for the attacker to automate the process. Many methods were developed to solve text CATPCHAs [47]–[49]. Beyond the classical optical character recognition (OCR) algorithms, researcher showed that attackers can abuse audio CAPTCHAs, which are often provided alongside classical CAPTCHA challenges to improve website accessibility [50]. As mentioned above, a MitM attack in the application layer can be applied to solve CAPTCHA challenges [38].

Security questions is another mechanism that has been studied. Previous research showed that many security questions are weak, either due to guessable responses (low entropy) or due to answers that are publicly available online [29]–[32]. These works also discuss ways to choose good security questions.

IX. Password Reset Process Auditing

Our work discovered vulnerabilities in the password reset process of the most popular websites in the world. If well-secured websites like Google and Facebook are vulnerable, it is reasonable to assume that many other websites that have not been surveyed are vulnerable as well.

The damage that can be caused to billions of accounts over many websites makes it necessary to create a relatively short list of possible problems and secure alternatives. In this section we bring such a list that can be used to audit and to secure password reset procedures in websites. The section begins with general guidelines and continues with instructions about the different challenges discussed in the paper.

A. General Guidelines

We bring here guidelines that should be applied to prevent PRMitM attacks. We do not bring known and basic principles like limiting the number of tries in inserting the reset code, or to cancel previous codes once a new code is required.

1) Password-reset messages (SMS, phone call, email) must include the sending website, clear explanation about the meaning of the code (password reset), and a warning to avoid giving this code to any person or website. However, even all of those elements might not be enough to prevent the attack.
2) In spite of the previous point, password reset using either SMS or phone call can be implemented securely. See examples in Sections VII-B2 and VII-C. Yet, in addition to those countermeasures, the following points should be considered.
3) For each supported language, the password reset messages (SMS, phone call, email) must be sent in that language.
4) Test your password reset process for every supported language separately.
5) Notify the user when a password reset request is sent, to both the email and the phone. If the password reset is done via the phone, this is even more critical. Email notification to email account that got compromised is useless.
6) The link or the code sent to reset the password should be valid only for short time period, e.g., $1 - 15$ minutes.
7) If there are several ways to reset the password for a user, automatically disable the less secure ones. If it is impossible to use a secure password reset process, contact the user in advance and offer her both to add information that can be used to reset her password securely and to disable the (only) insecure ways.
8) Require several details about the user before sending the password-reset message (SMS, phone call, email). This prevents the easy option for the attacker to launch the attack given only the phone number of the user, without knowing anything else about the user.

Finally, although the recommendations of this section are given mainly in the perspective of the PRMitM attack, it is important to note that according the NIST Digital Authentication Guideline, due to other security problems (stronger attacker model) it is not recommended to rely only on SMS or phone calls for authentication [51].

B. Security Questions

Avoid relying on security questions. Security questions are relatively easy to bypass, using either PRMitM attacks or other techniques [29]–[32].

What to do with users who do not have an alternative email account or a phone number. We offered two alternatives: (1) rely on email accounts of friends (Section VII-E), and (2) use security questions that are strongly related to the user's actions in the website (Section VII-A). The second option is still vulnerable to other attacks and hence, less preferred. When a user does not give a website another email address or phone number as alternatives, the website should at least warn the user about the dangers of relying on security questions, and encourage the user to move to the alternatives.

C. SMS Code

Specify the sender name. Use SMS spoofing to give indication about the real sender.

Do not send the code as clear text. Many users do not read the messages and just detect and copy the code. We offer an alternative: send SMS with detailed message and with a long link instead (Section VII-B2).

D. Phone Call

Add interactivity to the process to make sure that the users listen to the message and understand what they are doing. For example, after reading a detailed message, do not immediately give the code, but ask the user a few questions to make sure she understands the situation.

X. Conclusions

This paper introduced the PRMitM attack, which exploits a set of vulnerabilities in password reset procedures of popular (and other) websites and mobile applications. The attack allows a weak attacker to take over accounts of many websites, including Google and Facebook and other popular websites we surveyed. We evaluated the attacks and pointed at vulnerabilities and weaknesses of the password reset processes.

Although simple defense like more detailed SMS messages seems to be enough, our experiments indicate that this is not the case. We designed defenses and evaluated them compared to the existing implementations of Google and Facebook; our experiments show that our proposed defenses improve the security significantly. Finally, to help the many vulnerable websites to test and improve their password reset processes, we created a list of rules and recommendations for easy auditing.

Acknowledgments

The authors wish to express their gratitude to the Research Fund of the Research Authority of the College of Management Academic Studies, Rishon Lezion, Israel, for the financial support provided for this research.

References

[1] R. Morris and K. Thompson, "Password security: A case history," *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[2] Troy Hunt, "Have I Been Pwned?" https://haveibeenpwned.com/.

[3] S. Gaw and E. W. Felten, "Password management strategies for online accounts," in *Proceedings of the second symposium on Usable privacy and security*. ACM, 2006, pp. 44–55.

[4] X. de Carné de Carnavalet and M. Mannan, "From very weak to very strong: Analyzing password-strength meters," in *Network and Distributed System Security Symposium (NDSS 2014)*. Internet Society, 2014.

[5] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor, "Encountering stronger password requirements: user attitudes and behaviors," in *Proceedings of the Sixth Symposium on Usable Privacy and Security*. ACM, 2010, p. 2.

[6] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman, "Of passwords and people: measuring the effect of password-composition policies," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 2595–2604.

[7] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer *et al.*, "How does your password measure up? the effect of strength meters on password creation," in *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, 2012, pp. 65–80.

[8] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor, "Designing password policies for strength and usability," *ACM Transactions on Information and System Security (TISSEC)*, vol. 18, no. 4, p. 13, 2016.

[9] M. Zviran and W. J. Haga, "A comparison of password techniques for multilevel authentication mechanisms," *The Computer Journal*, vol. 36, no. 3, pp. 227–237, 1993.

[10] J. J. Yan, A. F. Blackwell, R. J. Anderson, and A. Grant, "Password memorability and security: Empirical results." *IEEE Security & privacy*, vol. 2, no. 5, pp. 25–31, 2004.

[11] L. V. Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using Hard AI Problems for Security," in *EUROCRYPT*. Springer-Verlag, 2003, pp. 294–311. [Online]. Available: http://dl.acm.org/citation.cfm?id=1766171.1766196

[12] M. L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, P. G. Kelley, R. Shay, and B. Ur, "Measuring password guessability for an entire university," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 173–186.

[13] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," *Communications of the ACM*, vol. 50, no. 10, pp. 94–100, 2007.

[14] J. M. Jeff Williams and N. Mattatall, "Cross Site Scripting Prevention Cheat Sheet," https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet, March 2016.

[15] Paul Petefish, Eric Sheridan, and Dave Wichers, "Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet," https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet, 2015.

[16] R. Hansen and J. Grossman, "Clickjacking," *Sec Theory, Internet Security*, 2008.

[17] S. Lekies, B. Stock, M. Wentzel, and M. Johns, "The unexpected dangers of dynamic javascript," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 723–735.

[18] N. Gelernter and A. Herzberg, "Tell me about yourself: The malicious captcha attack," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 999–1008.

[19] T. Van Goethem, W. Joosen, and N. Nikiforakis, "The clock is still ticking: Timing attacks in the modern web," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1382–1393.

[20] R. Dhamija and J. D. Tygar, "Why phishing works," in *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, 2006, pp. 581–590.

[21] Noriaki Hayashi, "New Phishing Technique Outfoxes Site Owners: Operation Huyao," http://blog.trendmicro.com/trendlabs-security-intelligence/new-phishing-technique-outfoxes-site-owners-operation-huyao/, November 2014.

[22] Jim Youll, "Fraud Vulnerabilities in SiteKey Security at Bank of America," http://cr-labs.com/publications/SiteKey-20060718.pdf, July 2006.

[23] Google, "Prevent phishing attacks on your users," https://support.google.com/a/answer/6197480?hl=en, November 2016.

[24] J. A. Chaudhry, S. A. Chaudhry, and R. G. Rittenhouse, "Phishing attacks and defenses," *International Journal of Security and Its Applications*, vol. 10, no. 1, pp. 247–256, 2016.

[25] A. Dvorkin and A. Herzberg, "Effective and usable browser-based defenses against phishing," *International Journal of Electronic Security and Digital Forensics*, 2009, accepted with revisions, revised version sent.

[26] S. Egelman, L. F. Cranor, and J. Hong, "You've been warned: an empirical study of the effectiveness of web browser phishing warnings," in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 1065–1074. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357219

[27] Alexa, "Top Sites," http://www.alexa.com/topsites, May 2016.

[28] J. Bonneau, E. Bursztein, I. Caron, R. Jackson, and M. Williamson, "Secrets, lies, and account recovery: lessons from the use of personal knowledge questions at google," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 141–150.

[29] M. Just, "Designing and evaluating challenge-question systems," *IEEE Security & Privacy*, no. 5, pp. 32–39, 2004.

[30] M. Jakobsson, E. Stolterman, S. Wetzel, and L. Yang, "Love and authentication," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 197–200.

[31] A. Rabkin, "Personal knowledge questions for fallback authentication: Security questions in the era of facebook," in *Proceedings of the 4th symposium on Usable privacy and security*. ACM, 2008, pp. 13–23.

[32] S. Schechter, A. B. Brush, and S. Egelman, "It's no secret. measuring the security and reliability of authentication via "secret" questions," in *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE, 2009, pp. 375–390.

[33] Facebook, "Facebook Login for your Apps & Websites," https://developers.facebook.com/products/login, May 2016.

[34] D. Irani, M. Balduzzi, D. Balzarotti, E. Kirda, and C. Pu, "Reverse social engineering attacks in online social networks," in *Detection of intrusions and malware, and vulnerability assessment*. Springer, 2011, pp. 55–74.

[35] Bill Marczak and John Scott-Railton, "The Million Dollar Dissident: NSO Group's iPhone Zero-Days used against a UAE Human Rights Defender," https://citizenlab.org/2016/08/

million-dollar-dissident-iphone-zero-day-nso-group-uae/, August 2016.

[36] Chris Smith, “One seemingly innocuous text message can wreck your Android phone’s security,” http://bgr.com/2016/02/16/android-sms-malware-attack-mazar-bot/, February 2016.

[37] S. Schechter, S. Egelman, and R. W. Reeder, “It’s not what you know, but who you know: a social approach to last-resort authentication,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 1983–1992.

[38] M. Egele, L. Bilge, E. Kirda, and C. Kruegel, “Captcha smuggling: hijacking web browsing sessions to create captcha farms,” in *Proceedings of the 2010 ACM Symposium on Applied Computing*. ACM, 2010, pp. 1865–1870.

[39] K. Thomas and D. M. Nicol, “The koobface botnet and the rise of social malware,” in *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on*. IEEE, 2010, pp. 63–70.

[40] T. Lauinger, V. Pankakoski, D. Balzarotti, and E. Kirda, “Honeybot, your man in the middle for automated social engineering.” in *LEET*, 2010.

[41] M. Huber, S. Kowalski, M. Nohlberg, and S. Tjoa, “Towards automating social engineering using social networking sites,” in *Computational Science and Engineering, 2009. CSE’09. International Conference on*, vol. 3. IEEE, 2009, pp. 117–124.

[42] T. Dougan and K. Curran, “Man in the browser attacks,” *International Journal of Ambient Computing and Intelligence (IJACI)*, vol. 4, no. 1, pp. 29–39, 2012.

[43] V. Roth, W. Polak, E. Rieffel, and T. Turner, “Simple and effective defense against evil twin access points,” in *Proceedings of the first ACM conference on Wireless network security*. ACM, 2008, pp. 220–235.

[44] Y. Song, C. Yang, and G. Gu, “Who is peeping at your passwords at starbucks?-to catch an evil twin access point.” in *DSN*, vol. 10, 2010, pp. 323–332.

[45] C. Karlof, J. D. Tygar, and D. Wagner, “Conditioned-safe ceremonies and a user study of an application to web authentication.” in *NDSS*, 2009.

[46] Brad Stone, “Breaking Google CAPTCHAs for Some Extra Cash,” http://bits.blogs.nytimes.com/2008/03/13/breaking-google-captchas-for-3-a-day/?_r=0, 2008. [Online]. Available: http://bits.blogs.nytimes.com/2008/03/13/breaking-google-captchas-for-3-a-day/?_r=0

[47] E. Bursztein, M. Martin, and J. Mitchell, “Text-based captcha strengths and weaknesses,” in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 125–138.

[48] C. Cruz-Perez, O. Starostenko, F. Uceda-Ponga, V. Alarcon-Aquino, and L. Reyes-Cabrera, “Breaking recaptchas with unpredictable collapse: heuristic character segmentation and recognition,” in *Pattern Recognition*. Springer, 2012, pp. 155–165.

[49] E. Bursztein, J. Aigrain, A. Moscicki, and J. C. Mitchell, “The end is nigh: Generic solving of text-based captchas,” in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, 2014.

[50] E. Bursztein and S. Bethard, “Decaptcha: breaking 75% of ebay audio captchas,” in *Proceedings of the 3rd USENIX conference on Offensive technologies*. USENIX Association, 2009, p. 8.

[51] Paul A. Grassi and James L. Fenton and Elaine M. Newton and Ray A. Perlner and Andrew R. Regenscheid and William E. Burr and Justin P. Richer and Naomi B. Lefkovitz and Jamie M. Danker and Yee-Yin Choong and Kristen K. Greene and Mary F. Theofanos, “DRAFT NIST Special Publication 800-63B: Digital Authentication Guideline,” https://pages.nist.gov/800-63-3/sp800-63b.html, November 2016.