Two-factor Authentication: Is the World Ready? Quantifying 2FA Adoption

Thanasis Petsas, Giorgos Tsirantonakis, Elias Athanasopoulos, Sotiris Ioannidis
Foundation for Research and Technology—Hellas (FORTH), Greece
{petsas, tsirant, elathan, sotiris}@ics.forth.gr

ABSTRACT

As text-based passwords continue to be the dominant form for user identification today, services try to protect their costumers by offering enhanced, and more secure, technologies for authentication. One of the most promising is *two-factor authentication* (2FA). 2FA raises the bar for the attacker significantly, however, it is still questionable if the technology can be realistically adopted by the majority of Internet users. In this paper, we attempt a first study for quantifying the adoption of 2FA in probably the largest existing provider, namely Google. For achieving this, we leverage the password-reminder process in a novel way for discovering if 2FA is enabled for a particular account, *without* annoying or affecting the account's owner. Our technique has many challenges to overcome, since it requires issuing massively thousands of password reminders. In order to remain below the radar, and therefore avoid solving CAPTCHAs or having our hosts blocked, we leverage distributed systems, such as TOR and PlanetLab. After examining over 100,000 Google accounts, we conclude that 2FA has not yet been adopted by more than 6.4% of the users. Last but not least, as a side-effect of our technique, we are also able to exfiltrate private information, which can be potentially used for malicious purposes. Thus, in this paper we additionally present important findings for raising concerns about privacy risks in designing password reminders.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*Authentication*

General Terms

Security; Measurement;

Keywords

Two-factor; Authentication; Adoption; Password Reminder; Privacy Leak;

EuroSec'15, April 21–24 2015, Bordeaux, France
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3479-2/15/04 ...$15.00.
http://dx.doi.org/10.1145/2751323.2751327.

1. INTRODUCTION

The wild adoption of on-line services by Internet users has raised substantially the need for better authentication. Users are now required to manage tens of different passwords. Although the capabilities of humans in memorizing secrets are still under research [12], it is evident that users have a hard time coping with the myriads of passwords they have to *always* remember. Using specialized software, commonly known as *password managers* [26, 31], might be a solution, however, researchers have concluded that, in certain cases, password managers can make things worse [16]. Today, it is a common practice for users to select a few passwords that are recycled from service to service [21]. Recycling of passwords [22, 32] may have severe consequences [15] when there is a password leak [29]. Therefore, services have started to employ server-side countermeasures for fighting against password stealing. One of them, and perhaps the most well known, is *two-factor authentication* (2FA).

2FA is an enhanced authentication mechanism for protecting users that have their password stolen either by leaking a server's password database [29] or *phishing* [19]. An attacker that owns the victim's password needs to have access to an additional communication channel for receiving a one-time-generated token, which should accompany the password during authentication. The proliferation of cellphone usage has significantly assisted 2FA deployment. The user can provide the service with her phone number, and the service can communicate the 2FA token to the user's cell during authentication. Therefore, the attacker that owns the user's password has to also compromise the user's cellphone.

Although 2FA raises the bar for the attacker significantly, it is still questionable if the technology can be realistically adopted by the majority of Internet users. There are still many open concerns regarding scalability and usability. For example, it is unclear if users can utilize 2FA with every site they communicate, how conveniently migration can take place when a user is changing her phone number or a user is losing her cellphone, or how easy is to supply the token upon every authentication session. In this paper, we attempt to experimentally quantify 2FA adoption in the wild. It is important to stress that there are so far unofficial studies estimating 2FA adoption in a range of 2%-5% [30] [1], but the experimental base of these results is still unpublished and unknown.

For quantifying 2FA adoption we follow a novel approach, based on leveraging the password-reminder procedure designed by one of the main service providers that has enabled 2FA, namely Google. We are targeting Google on purpose, since according to them, 2FA

[1] Discussion with Paul Moore related to this study was held over private communication in October 2014. We are still not aware of the scientific methodologies used for concluding these estimates about 2FA adoption.

adoption seems really rapid: *"Our experience with 2-Step Verification (2SV) has been good. Adopted by millions, it's among the largest two-factor authentication deployments in the world. Nearly a quarter million accounts added 2SV during the two days after Mat Honan's story broke, illustrating a phenomenon that we observe more broadly: people take security more seriously after an acquaintance or public figure has suffered harm."* [24]. Hence, our findings are based on analyzing a successful 2FA deployment.

1.1 Contributions

Our findings are sound and indicative. First, by analyzing more than 100,000 Google accounts we are able to project that 2FA has not been adopted by more than 6.4% of users, yet. Second, our technique unveils a series of privacy concerns related to personal information leakage while a password-reminder process takes place. Third, and finally, we conclude with suggestions for improving password reminders.

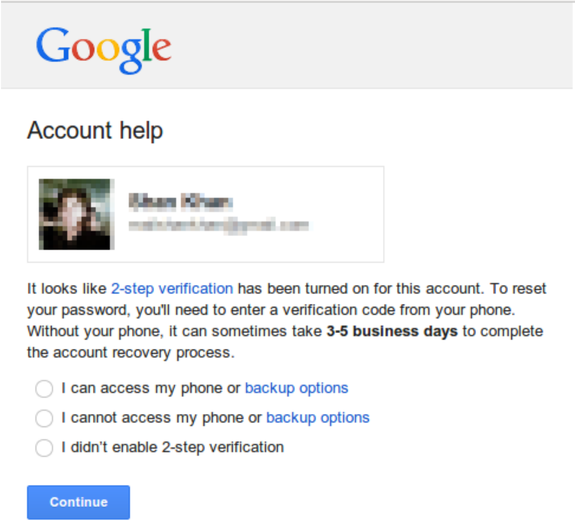
2. MEASURING 2FA IN THE WILD

2.1 Who is Using 2FA?

Two-factor authentication is widely used in online banking and increasingly adopted by many Internet service providers. One of the key questions we attempt to answer is the following: *"In what degree are 2FA schemes used by current service providers?"* In order to understand in what extent 2FA is being deployed by services and web sites in recent years, we attempt to find official announcements confirming that a service has started to support 2FA. We collected many such announcements and extracted the date they were published which we used as an indication of when a service started supporting 2FA. Most of the needed information is derived from a publicly available archive we found, that provides a list of websites that support 2FA [7]. Figure 1 shows the increasingly adoption of 2FA by various service providers through time, as reflected from the official announcements we collected. We see that, as the time goes by, an increasing number of services begins to use 2FA. As popularity of 2FA is growing, we expect to see more services deploying it in the coming years. Moreover, we see that Google, along with Facebook and Yahoo, was one of the first service providers that introduced a 2FA method [9]. Since, more and more companies provide 2FA schemes to improve security of their users, it is important to know if users are willing to use this new kind of technology. In other words, the main question we try to answer in this study is *"What is the adoption level of 2FA by the end users?"* To do this, we attempt to quantify the adoption of 2FA in Google provider. We believe that Google is the ideal choice for such a study and can help us draw general conclusions about 2FA adoption, due to two key elements: (i) It is one of the largest existing providers in terms of users, (ii) It probably contains the largest percentage of users that are aware of 2FA (since it is one of the first service providers started using it).

2.2 Google Password Reminder

Google lets its users access various services such as Gmail, Google Plus, Google Maps, YouTube etc., by using a single account. A Google account comprises of an email address (usually created through Gmail or from another provider), and a password. During the creation of a Google account, the user is strongly encouraged (but not enforced) to provide a secondary (recovery) email address, or a valid mobile phone number for future verification of her account. This can be used in order to be able to recover her password, in case she has forgot it or her account has been hijacked. Google also provides a 2FA option, as already mentioned, which is named as *2-Step Verification* (2SV), and is used as an additional security measure against account hijacking. 2SV requires a validation code upon a user login which can be generated trough various ways, e.g., by a mobile app, called *Google Authenticator*, through an SMS text message sent by Google, or even through a voice call [23]. Recently, Google announced *Security Key*, which is a physical USB device, used as a second factor, to verify an account login by inserting it in a computer; simplifying in this way the two-factor authentication process [10]. Furthermore, Google provides a password-reminder option, in order for a user to be able to recover her password, if she has forgotten it, or her account has been hijacked. We found that the different verification methods, a user enabled during her account generation, are reflected in the password-reminder process (Figure 2), through a sequence of actions.This procedure is presented in detail in Section 3.



Account help

[redacted]

It looks like 2-step verification has been turned on for this account. To reset your password, you'll need to enter a verification code from your phone. Without your phone, it can sometimes take **3-5 business days** to complete the account recovery process.

○ I can access my phone or backup options

○ I cannot access my phone or backup options

○ I didn't enable 2-step verification

Figure 2: *A screenshot of Google's password reminder facility, revealing that the 2FA is enabled for a user in our dataset.*

2.3 Challenges

The main challenges of our study are based on the fact that all measurements should be carried out without being intrusive to the target web service, i.e., Google. Many sites deploy CAPTCHA [8] in order to prevent crawlers, and other automated programs, from accessing their web content. As expected, Google also follows this approach for its password-reminder facility. We found that a single IP address can issue up to six consecutive password-reminder requests on a given day without having to solve a CAPTCHA puzzle, regardless of whether the requests target the same Google account or not. This behavior limits the number of Google accounts that can be analyzed per day. This issue can be addressed by using a crowdsourcing marketplace service, such as Death by Captcha [3], Rumola [6], or Amazon's Mechanical Turk [1], which have been used in similar studies (for example, PlayDrone [33] crawls Google Play and downloads all the apps on a daily basis). We believe that these approaches most likely violate Google's terms of use, as well as the terms of other similar service providers and are ethically questionable. For our study, we chose to follow a distributed approach and limit our requests to a certain number per day, in order not to be intrusive. Furthermore, we take into account that a reasonable random sample of Google users is enough for projecting an estimation of 2FA adoption. We describe our methodology along with ethical considerations in Section 3.

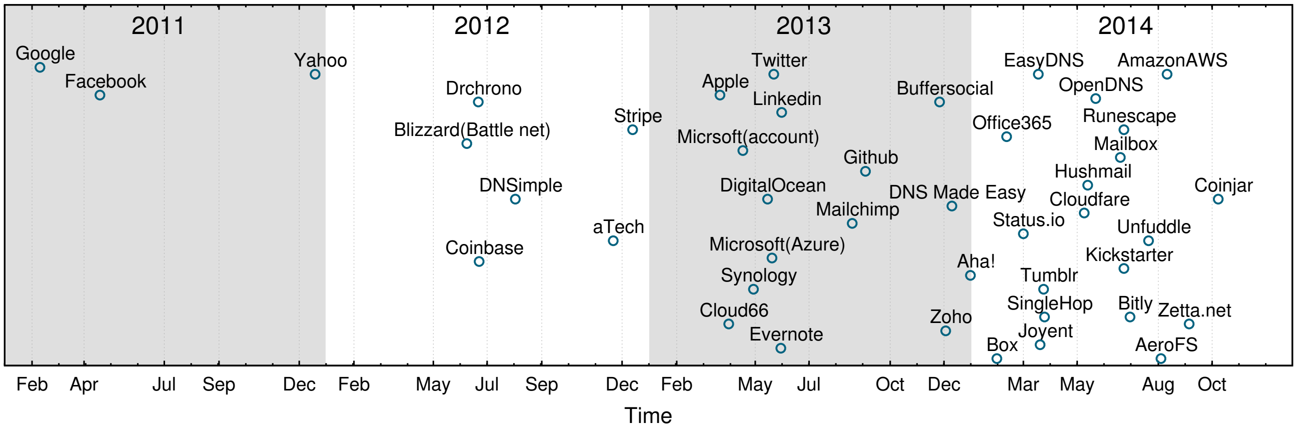


Figure 1: *2FA adoption by websites and web services through time.* 2FA technology has been gaining an increasingly popularity in recent years. We see that in 2011, only 3 service providers joined the 2FA technology, where in 2014 this number increased to 19. These results are taken from official announcements published by the services online.

3. METHODOLOGY

3.1 Dataset

In order to perform our measurements we needed a list of valid emails associated with Google accounts. According to Forbes [27], on the 9th of September 2014, almost 5 million Gmail addresses paired with passwords were leaked, and published online on a Bitcoin Security forum. The list contains only the Gmail addresses and was published in order for users to be able to check if their accounts fall into the set, taking the appropriate actions. We used this dataset in order to quantify the 2FA adoption of the users of Google provider. [2] Google, as well as other websites, that announced about the leaked Google accounts, suggested the users to enable 2FA for an extra layer of security [11]. Therefore, we expect that the rate of 2FA verification method, among the users in our dataset, to be increased.

3.2 Password Reminder Requests to Google

In this section, we describe in detail how our collection system works. We present, how we managed to issue a large number of password reminder requests to Google in a daily manner without being intrusive, the information we collect for each account in our dataset, and how we improved our system in order to reduce the number of the failed requests (i.e., those requiring CAPTCHA validation), to the smallest amount possible.

Collection System. As already mentioned in Section 2.2, to retrieve the verification methods of a user, there is a specific number of steps in Google's password-reminder interface that should be followed. This process is created to be carried by people that use a browser, and any attempt performed by automated means could be considered suspicious by Google. In order to issue password-reminder requests that appear as being performed by a real user, we used CasperJS [2], a browser automation utility that can execute navigation scenarios on a headless browser. The whole process is written in a CasperJS script which is called from a Python script. The Python script connects to an SQLite database (containing our dataset) and, selects randomly a number of not yet analyzed email addresses which provides as input, one by one, to CasperJS. The four steps required to analyze a given Google account are depicted in the Figure 3. In step ❶, our host visits the Google Sign in page and clicks the *Need help?* link, found at the bottom of the sign-in form. This link activates the password-reminder facility and forwards the user to another page entitled *"Having trouble signing in?"*. In step ❷, our host provides an email address from our dataset as input, it checks the option *I don't know my password*, and clicks the button *continue*. If the account is invalid (e.g., nonexistent, disabled or deleted), then, the process will return this result and our host will end its discovery procedure there. Otherwise, it will be forwarded to a third page named *Account Help*. In this page, the user is prompted to enter the last password she remembers in order to continue the process. In step ❸, our host clicks the button *I don't know*, implying that is not aware of any passwords, and then in the current page a new piece of content will be automatically generated with the verification option of the user (step ❹). Then, we parse the HTML content for specific strings, implying the verification method that Google suggested to the user, in order to recover her password. The verification method is stored in our database, and the specific email is marked as analyzed. We found all the possible generated strings that Google uses in password-reminder process and incorporated them to our scripts. To do so, we created various Google accounts with different combinations of verification methods covering every possible case, and issued a number of password-reminder requests. During the password-reminder process, Google suggests one of the user enabled verification methods with a priority. What follows are the different verification options in order of appearance (organized from high to low priority), along with the corresponding generated strings: 2SV verification method (*"It looks like 2-Step Verification has been turned on"*), Mobile phone verification (*"Enter your phone number"*), Recovery mail verification (*"Confirm access to my recovery email"*) and No verification (*"select one of the options below to reset your password"*). This means that if a user has enabled both the 2SV and recovery email verification methods, only 2SV will be displayed in the last step of the password-reminder process (step ❹). In some cases, no results were retrieved by our scraping due to an erroneous account, that is an invalid Google account due to typos in email address [3], a disabled account or a deleted one. We were able to identify this by the generated messages.

[2] The dataset used in this study is available at `https://forum.bits.media/uploads/manual_09_2014/google_5000000.7z`

[3] We excluded from our dataset all the email addresses that were not valid Gmail addresses, or duplicates (2.53% of our dataset).

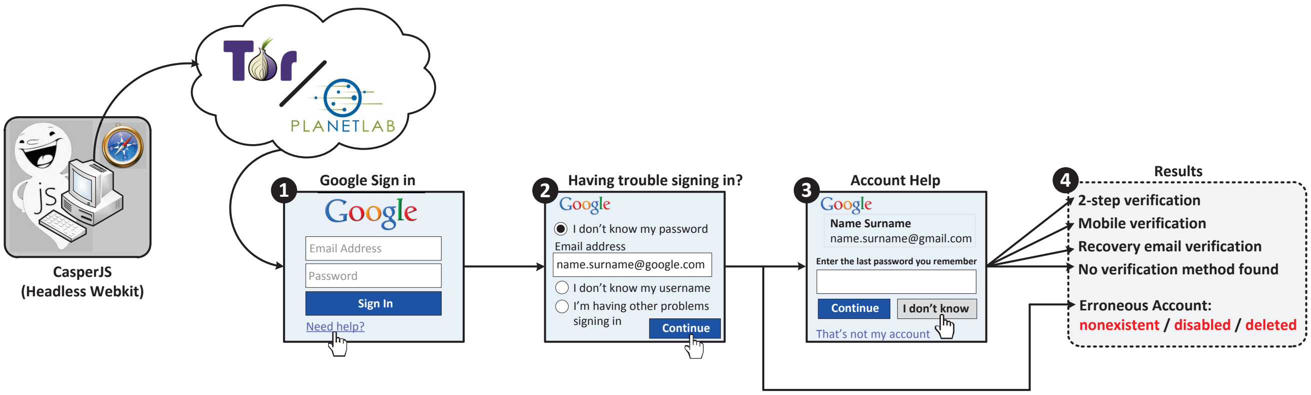
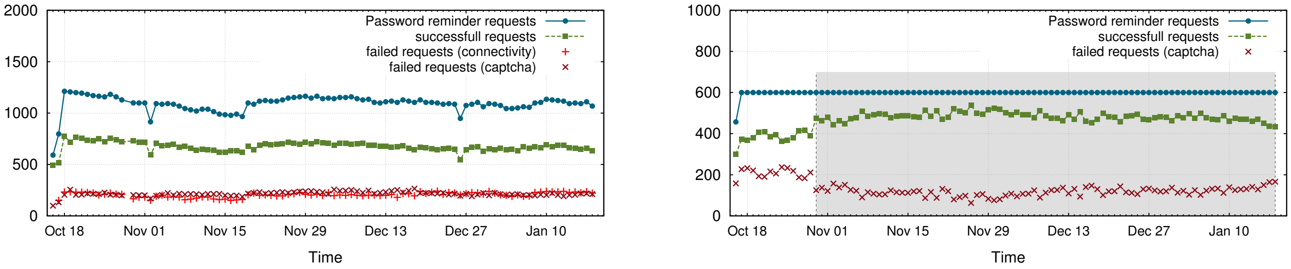


Figure 3: *Overview of our methodology*: We connect to Google, using a headless browser proxying our connections through TOR and PlanetLab. We provide an email to Google's sign on page, and through 3 steps (clicks), we are able to infer if this account uses 2FA or not.



(a) PlanetLab

(b) TOR

Figure 4: *Password-reminder requests per day through PlanetLab and TOR.* In (b), the light gray rectangle contains the requests send by our updated TOR client that chooses a new TOR node for each request. As we see, the amount of CAPTCHAs has been reduced substantially.

Browser Enhances. In order each password-reminder request to be more close to one performed by a separate real user, we add the following attributes, to our system: *Cookies:* We clear all cookies from our headless browser for each email address verification discovery. *User Agent:* We use a random user agent among 80 most common ones used in the wild [5]. *Delay:* We randomized the amount of time our script should wait before perform consecutive actions (e.g., the clicks pages it visits). *Fake Referer:* Before starting the password recovery, we visit a random search engine (e.g., Bing, Yahoo, etc.) pointing to Gmail help page, in order our landing in Google to appear as if it was through a search engine.

Distributed Approach. In order to avoid restrictions imposed by Google (e.g., requests limit per IP address, CAPTCHAs), we used various nodes from two popular distributed networks as proxies (Figure 3). The first one is TOR [20], which is a network of hosts around the world helping users to surf the Internet anonymously. The other one is PlanetLab [17], that is a group of computer hosts, shared by academic institutions globally, available as a testbed for performing distributed experiments. Each password-reminder request (i.e., the 4 steps until extracting the verification results) was proxied through a new TOR circuit (via sending a NEWNYM signal to our TOR client). We noticed that, a portion of the TOR exit nodes were the same across many generated circuits. In order to bypass this restriction, we instructed our TOR client to use a new generated circuit, only if its exit node was totally new compared to the previous ones. To do this, we changed our scraper script to visit `whatsmyip.org` after each request, in order to retrieve the IP address of the current TOR exit node and append it to the `ExcludeExitNodes` list of the TOR configuration file (`torrc`). We then send a HUP signal to the TOR client in order to load the updated file. After this change, the CAPTCHA rate was dramatically decreased by 19.8% compared with the default configuration.

System Deployment. We deployed our scraper scripts to use both TOR and PlanetLab nodes. We set a *limit* of 1800 password-reminder requests per day (600 requests through TOR and up to 1200 requests through PlanetLab). We do not use threads to issue our requests in order not to be intrusive. The total number of password-reminder requests we issue per day is shown in Figure 4. We perform, on average, 1,683 requests per day, 1,137 of which are successful and 546 result in CAPTCHAs (and other failures due to connectivity issues with PlanetLab nodes). In Figure 4 (gray rectangle area), we see that, when using our new TOR client configuration, which chooses a new unused TOR exit node for each request, the amount of CAPTCHAs is reduced.

3.3 Ethics

In our study, we analyze over 100 thousands of Google accounts in order to discover which of them have enabled 2FA. Before starting our study, we first confirmed that our approach does not annoy or affect the owner of the account in any case. We did this by replicating our experiments on various Google accounts created for this purpose. We found out that our approach terminates the process soon, and, therefore, the user is never notified as if a failed password-reminder has occurred on her account. Moreover, if a request for a specific account results in a CAPTCHA, we do not perform the same request during the course of the day. In order not to be intrusive, we apply a limit of 1800 password-reminder requests per day. According to [14], password reset rates have been estimated as one recovery per every four users per month. Since the number of active registered users in Google is over 425 million [4], Google receives over 106 million password reminder requests per month, that is over 3.5 million requests per day. Thus, the number of password-reminder requests we issue per day is less than 0.005% of the total daily requests it receives, and in any case can not be considered as intrusive. Furthermore, as far as the collecting data is concerned, we *do not store any privacy related information* of the user (e.g., images, phone numbers etc.); we only store aggregated data obtained by increasing specific counters.

4. RESULTS

2FA Adoption. We analyzed 110,007 accounts in total, of which 101,047 (91.9%) were valid accounts (i.e., we collected the verification methods of the corresponding users), and 8,960 (8.14%) were erroneous. Table 1 summarizes the results of our study. Interestingly, we see that *only 6.39% of the analyzed accounts have enabled the 2-Step Verification method.* This shows that 2FA has not been adopted on a large scale yet. Although the fact that smartphones are steadily gaining popularity globally and are used by million of people around the world, 2FA, which is based on these devices, does not seem to share a similar growth. We found this result by analyzing the user accounts from one of the first providers that deployed 2FA in their services. We believe that this percentage would be even lower in other providers. Furthermore, we found that 61.75% of the users have enabled mobile phone verification. We see that, although more than the half of the users (i.e., 68.14% – those who use mobile verification plus those who use 2SV) have registered their mobile phone to Google for account verification, only 9.38% of them are enrolled with the 2FA feature. As far as the rest of the users are concerned, we found that 17.03% of them uses only recovery email as a verification method, while 14.83% of them have not enabled any of the verification methods. These users, in order to recover their account, they have to submit a form to Google answering a series of personal questions.

Erroneous Accounts. Table 2 lists the accounts that did not return verification data. By inspecting the received responses, we found that almost half of those accounts (50.83%) were deleted by the users. Even though we are not able to know the reasons of this fact, this may be because those users were alarmed when they learned that their account had leaked to third parties. Moreover, we found that 103 accounts were disabled by Google, for reasons unknown to us. The rest of erroneous accounts (48.02%) did not exist at all.

2FA Sustainability. After gathering a sufficient number of accounts (6,225) that have 2FA enabled, we tried to measure the sustainability of 2FA through time as reflected from our sample. As mentioned previously, we store aggregated data by increasing counters. We also store a hash for each account with 2FA enabled. So, in order to perform this measurement, we needed only to recheck the accounts that their hash fell into the 2FA set to verify if they still use 2FA. Table 3, summarizes the results of this analysis. As our accounts were first analyzed at different times during our measurements, we present the results for different time intervals (*recheck interval*) between the first and the second check. Overall, we observe that only 2% of the accounts disabled 2SV. As, we see, this portion increases with time; e.g., for accounts that were rechecked after at least 3 months, this portion is equal to 2.77%. Although we are not able to know why this happens, we speculate that this is due to usability issues these users experience.

User Data Exposed. While this study was carried out, we found that password reminder of Google can expose sensitive information, during the third step of the password-reminder process (Figure 3, step ❸). The information exposed in this step, contains the name and surname of the user along with a photograph of herself (if set previously by her), as shown in Figure 2. We found that 58,761 out of 101,047 valid accounts we analyzed, that is 58.15%, contained a photograph of the user. In addition, in step 4 of the process (Figure 3, step ❹), if a mobile verification method is selected, the 3 last numbers of the mobile phone of the users appear in plaintext (and the rest of them are hidden using a number of '*' characters). Furthermore, in the same step, if an email verification method is chosen, the first 3 characters of the secondary email are displayed (and the rest of them are covered by '*' too). Thus, apart from the 3 characters that are fully exposed, the exact size of email address and mobile number are exposed too. Although the severeness of this exposure is not yet assessed, in the past, it has been documented that partial information disclosure can be combined with other personal data for performing a really devastating attack [28].

5. DISCUSSION

5.1 Password Reminder Facilities Across Different Services

In order to obtain a better view of the information exposed by a typical password-reminder facility, we studied various of them deployed in the wild. We selected the following popular service providers: Microsoft, Yahoo, Apple, Facebook, Twitter, LinkedIn, Github, Dropbox and Evernote. We would like to gain insights on whether their password-reminder process is similar to the one used by Google and what kind of information is revealed by each of them. To do so, we created multiple accounts for each service, some of which had 2FA enabled and some that did not, and attempted to see if verification methods are reflected by the corresponding password reminder facilities. Table 4 summarizes the results of the comparison. As we see, all the services except from Apple and Microsoft do not expose information whether 2FA is enabled for a specific account. Surprisingly, we found that during the password-reset process of Apple, if you provide the Apple ID of a user, it is revealed if a user has 2FA enabled. This means that it would be feasible to replicate our study there. Regarding Microsoft, we found that 2FA information is exposed only if a user has installed an authenticator app in her phone; otherwise, no information is revealed. Nonetheless, we found that a CAPTCHA is required to be solved in order to get this information. in every password-reminder process attempt. Thus, we would not be able to perform the same study to Microsoft.

5.2 Improving Google's Password Reminder Facility

Based on the features of the studied password-reminder facilities, we suggest various measures that Google may deploy in order to improve its own password-reminder process. One first measure could be the use of CAPTCHA in every password-reminder attempt, as Microsoft already does. This would hinder crawlers from easily harvesting such information. Moreover, sensitive information related to the user, that is name, surname and users photographs should not be exposed during the password-reminder process. In addition, the 2FA option during the password-reset process does not seem to be necessary, since the recovery email and mobile phone verification methods are provided. Solutions like the one followed by Twitter could also be adopted, where user

Verification Method	Google Accounts Total	 Percentage
2-Step Verification (2SV)	6,455	6.39%
Mobile phone	62,396	61.75%
Recovery email	17,207	17.03%
No verification	14,988	14.83%
Total	101,047	100%

Table 1: *Verification methods used by Google accounts we analyzed.* We see that only 6.39% of the users use 2-Step Verification.

Type	Erroneous Accounts Total	 Percentage
Deleted	4,554	50.83%
Nonexisted	4,303	48.02%
Disabled	103	1.15%
Total	8,960	100%

Table 2: *Erroneous Google accounts that returned not valuable results.*

	2SV Accounts		
Recheck Interval	**Total**	**Stayed**	**Abort**
any (all accounts)	6,225	6,110	115 (1.85%)
≥ half month	5,773	5,658	115 (1.99%)
≥ 1 month	4,739	4,633	106 (2.24%)
≥ 2 months	2,697	2,626	71 (2.63%)
≥ 3 months	830	807	23 (2.77%)

Table 3: *Sustainability of 2SV through time.* We see that a small portion of users abandon using 2SV after a period of time.

	Password Reminder Information	
Service Provider	**URL**	**Is 2FA Exposed?**
Microsoft	`https://account.live.com/ResetPassword.aspx`	✓*
Yahoo	`https://edit.europe.yahoo.com/forgot`	✗
Apple	`https://iforgot.apple.com/password/verify/appleid`	✓
Facebook	`https://www.facebook.com/login/identify?ctx=recover`	✗
Twitter	`https://twitter.com/account/begin_password_reset`	✗
LinkedIn	`https://www.linkedin.com/uas/request-password-reset`	✗
Github	`https://github.com/password_reset`	✗
Dropbox	`https://www.dropbox.com/forgot`	✗
Evernote	`https://www.evernote.com/ForgotPassword.action`	✗
Google	`https://www.google.com/accounts/recovery`	✓

Table 4: *Comparison of password-reminder features across different service providers.* *Microsoft reveals if a user has 2FA enabled for those that use a mobile app available for this purpose.

starts the password reminder from the Twitter website and the whole password-reminder process continues through email (i.e., a different communication channel), so the information cannot be leaked to a third party.

6. RELATED WORK

2FA Studies. Weir et al. [34] performed a user case study asking e-banking customers to rate different 2FA methods in terms of security, quality and convenience. Overall, they found that users prefer usability above all and did not see the need for additional security. In a similar study [25] Ganson et al. asked mobile banking users to rate a single-factor and two 2FA schemes for telephone banking. They found that the average user took 20 more seconds to complete each 2FA process than the single-factor one, and 2FA appears to users as a more secure solution but less easy-to-use. In [18] De Cristofaro et al. asked by various 2FA-familiar users to rate the usability of the three most popular 2FA solutions with different forms, that is, email or SMS sent to the user, a mobile app used as authenticator and a hardware token that produces OTP codes. They observed that people who use 2FA for work prefer the mobile app option, while those who use it for personal and financial reasons prefer the text method (i.e., text code send through email or SMS). When these studies evaluated whether users are willing to use 2FA for authentication based on a limited number of users (tens to hundreds), we measure the adoption of 2FA on a large service provider by analyzing over 100,000 user accounts.

Password Reminder Studies. In [13], Bortz et al. show two different types of attacks that can leak private information from websites based on the time a web site takes to respond to HTTP requests. They also observe that this piece of information can be easily obtained by an attacker through the "Forgot my password" page of a site, without using their attacks. In our study, we noticed that, seven years after this work, web services, such as those of Google, still reflect private information of their users (e.g., name, surname or photographs) during the password-reset process. Gaw et al. [22], attempted to quantify password reuse of people across multiple online accounts and the different methods used for storing passwords. They found that after person's memory, the most commonly used technology for this purpose is password reminders. This means that password reminders is a significant aid people rely on to manage their passwords. In our study, we found that this piece of software can leak private information of the users and we proposed how it can be improved in terms of privacy.

7. CONCLUSION

In this paper we carried out the first experimental study for quantifying the adoption of two-factor authentication (2FA) by users. For this, we analyzed more than 100,000 Google accounts for identifying if 2FA is enabled or not. Our study projects that, so far, no more than 6.4% of users use 2FA. All of our measurements leveraged the password-reminder process of Google. Through out this paper we have identified and discussed various privacy concerns related to password reminders. More precisely, we have demonstrated that an attacker can easily exfiltrate sensitive information by just selecting the appropriate steps during the password-reminder process for a given account. Having pointed out the vulnerabilities of Google's password reminder, we proceeded and identified various properties that could in principle protect it. We believe that this paper (a) establishes experimentally – for the first time – the level of 2FA adoption by Internet users, and (b) highlights a number of privacy concerns for the design of password reminders.

Acknowledgements

This research was performed with the financial support of the Prevention of and Fight against Crime Programme of the European Commission – Directorate-General Home Affairs (project GCC). This work was also supported in part by the FP7 projects NECOMA, OPTET, funded by the European Commission under Grant Agreements No. 608533 and No. 317631.

8. REFERENCES

[1] Amazon mechanical turk. https://www.mturk.com/.

[2] Casperjs. http://casperjs.org/.

[3] Death by captcha. http://www.deathbycaptcha.com.

[4] Gmail now has 425 million active users. http://www.theverge.com/2012/6/28/3123643/gmail-425-million-total-users.

[5] Most common user agents. http://techblog.willshouse.com/2012/01/03/most-common-user-agents/.

[6] Rumola. http://skipinput.com/.

[7] Two factor auth (2fa) – list of websites and whether or not they support 2fa. https://twofactorauth.org/.

[8] Ahn, L. V., Blum, M., Hopper, N. J., and Langford, J. Captcha: Using hard ai problems for security. In *European Cryptology Conference (EUROCRYPT)* (2003).

[9] Blog, G. O. Advanced sign-in security for your google account. http://googleblog.blogspot.gr/2011/02/advanced-sign-in-security-for-your.html.

[10] Blog, G. O. S. Strengthening 2-step verification with security key. http://googleonlinesecurity.blogspot.gr/2014/10/strengthening-2-step-verification-with.html.

[11] Blog, O. O. S. Cleaning up after password dumps. http://googleonlinesecurity.blogspot.gr/2014/09/cleaning-up-after-password-dumps.html.

[12] Bonneau, J., and Schechter, S. Towards reliable storage of 56-bit secrets in human memory. In *USENIX Security Symposium* (2014).

[13] Bortz, A., and Boneh, D. Exposing private information by timing web applications. In *International Conference on World wide web (WWW)* (2007).

[14] Brostoff, S., and Sasse, A. M. Are passfaces more usable than passwords? In *International Conference on Human-Computer Interaction (HCI)* (2000).

[15] Cheswick, W. Rethinking passwords. *Communications of the ACM 56*, 2 (2013).

[16] Chiasson, S., van Oorschot, P. C., and Biddle, R. A usability study and critique of two password managers. In *USENIX Security Symposium* (2006).

[17] Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., and Bowman, M. Planetlab: An overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review (CCR) 33*, 3 (2003).

[18] De Cristofaro, E., Du, H., Freudiger, J., and Norcie, G. A comparative usability study of two-factor authentication. In *Proceedings of the Workshop on Usable Security (USEC)* (2014).

[19] Dhamija, R., Tygar, J., and Hearst, M. Why phishing works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI)* (2006).

[20] Dingledine, R., Mathewson, N., and Syverson, P. Tor: The second-generation onion router. In *USENIX Security Symposium* (2004).

[21] Florencio, D., and Herley, C. A large-scale study of web password habits. In *International Conference on World Wide Web (WWW)* (2007).

[22] Gaw, S., and Felten, E. W. Password management strategies for online accounts. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)* (2006).

[23] Google. About 2-step verification. https://support.google.com/accounts/answer/180744.

[24] Grosse, E., and Upadhyay, M. Authentication at scale. *IEEE Security and Privacy 11* (2013), 15–22. http://www.computer.org/cms/Computer.org/ComputingNow/pdfs/AuthenticationAtScale.pdf.

[25] Gunson, N., Marshall, D., Morton, H., and Jack, M. A. User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking. *Computers & Security* (2011).

[26] Halderman, J. A., Waters, B., and Felten, E. W. A convenient method for securely managing passwords. In *International Conference on World Wide Web (WWW)* (2005).

[27] Hill, K. Google says not to worry about 5 million gmail passwords leaked. http://www.forbes.com/sites/kashmirhill/2014/09/11/google-says-not-to-worry-about-5-million-gmail-passwords-leaked/.

[28] Honan, M. How apple and amazon security flaws led to my epic hacking. Wired Magazine, August 2012. http://www.wired.com/gadgetlab/2012/08/apple-amazon-mat-honan-hacking/.

[29] Kontaxis, G., Athanasopoulos, E., Portokalidis, G., and Keromytis, A. D. Sauth: Protecting user accounts from password database leaks. In *ACM Conference on Computer and Communications Security (CCS)* (2013).

[30] Paul Moore. Does two factor authentication actually weaken security? https://ramblingrant.co.uk/does-two-factor-authentication-actually-weaken-security/ (and private communication in October 2014).

[31] Ross, B., Jackson, C., Miyake, N., Boneh, D., and Mitchell, J. C. Stronger password authentication using browser extensions. In *USENIX Security Symposium* (2005).

[32] Shay, R., Komanduri, S., Kelley, P. G., Leon, P. G., Mazurek, M. L., Bauer, L., Christin, N., and Cranor, L. F. Encountering stronger password requirements: user attitudes and behaviors. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)* (2010).

[33] Viennot, N., Garcia, E., and Nieh, J. A measurement study of google play. In *ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)* (2014).

[34] Weir, C. S., Douglas, G., Richardson, T., and Jack, M. A. Usable security: User preferences for authentication methods in ebanking and the effects of experience. *Interacting with Computers* (2010).