Understanding Account Recovery in the Wild and its Security Implications

Yue Li, Zeyu Chen, Haining Wang, Kun Sun, and Sushil Jajodia

Abstract—Account recovery (usually through a password reset) on many websites has mainly relied on accessibility to a registered email, due to its favorable deployability and usability. However, it makes a user's online accounts vulnerable to a single point of failure when the registered email account is compromised. While previous research focuses on strengthening user passwords, the security risk imposed by email-based password recovery has not yet been well studied. In this article, we first conduct a measurement study to characterize the password recovery activities in the wild. Specifically, we examine the authentication and password recovery protocols from 239 traffic-heavy websites, confirming that most of them use emails for password recovery. We further scrutinize the security policy of leading email service providers and show that a significant portion of them takes no or marginal effort to protect user email accounts, leaving compromised email accounts readily available for mounting password recovery attacks. Then, we conduct case studies to assess potential losses caused by such attacks. Finally, we propose and implement a lightweight email security enhancement called Secure Email Account Recovery (SEAR) to defend against password recovery attacks by adding an extra layer of protection to password recovery emails.

Index Terms—Password management, authentication, secure communications

1 Introduction

Text-based passwords have been used as a dominating solution of user authentication for many decades [1], due to their favorable usability and the fact that they cannot be entirely replaced by other authentication approaches in the foreseeable future [2], [3]. Since text-based passwords are vulnerable to cracking and theft attacks [4], [5], significant research efforts have been made toward enhancing password security from different aspects, including measurement [6], [7], [8], [9], password policy [10], [11], password meters [12], [13], [14], [15], and password managers [16].

Whereas it is critical to secure a password at its creation and input procedures, account recovery as an important component in the entire framework of password-based authentication has been largely overlooked. Account recovery is an irreplaceable link in the password authentication chain. Not being able to provide an easy way to recover the password can cause user frustration, human labor waste, or even user loss. Meanwhile, the account recovery process should also be carefully designed to avoid backdoor threats. Today, most websites rely on accessibility of a registered email of a user to recover or reset forgotten passwords. Though email-based recovery is deployable, compatible, and easy to use, its security implication is understudied. A compromised email account could inevitably become a single-point-of-failure, since an attacker can easily reset the passwords of a victim's other online accounts. Note that such an account recovery attack can naturally circumvent security enhancements on passwords and directly compromise a large number of user accounts by resetting their passwords.

A simple and effective idea is to keep the email account safe. However, this does not happen in a practical world. There is a large number of email accounts leaking to malicious attackers. For example, it was suggested by a security firm in May 2016 [17] that more than 200 million email username/password combinations are in possession of hackers. Major email service providers including Gmail, Hotmail, Yahoo, and Mail.ru are all affected, and millions of email account credentials are compromised. Thus, it is important to understand the security implications of email-based account recovery. A systematic study on its vulnerability, potential damage, and defense has yet to be conducted.

In this paper, we first quantitatively measure the vulnerability of most websites to an account recovery attack. In particular, we manually investigate the account recovery protocols and authentication schemes adopted by the Alexa top 500 websites. We observe that 92.5 percent of the web services we examined rely on emails to reset user passwords, and in 81.1 percent of websites, their user accounts can be compromised by solely accessing the registered emails. The difference of 11.4 percent is due to the lack of username knowledge (i.e., the username/password credential is incomplete) or classifier-based authentication, where abnormal login attempts will be blocked. Afterward, we

- *Y. Li is with the Department of Computer Science, College of William and Mary, Williamsburg, VA 23185 USA. E-mail: yli@cs.wm.edu.*
- *Z. Chen is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA. E-mail: zeyuchen@udel.edu.*
- *H. Wang is with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA. E-mail: hnw@vt.edu.*
- *K. Sun and S. Jajodia are with the Department of Information Sciences and Technology, George Mason University, Fairfax, VA 22030 USA. E-mail: {ksun3, jajodia}@gmu.edu.*

Manuscript received 3 Dec. 2018; revised 5 Feb. 2020; accepted 11 Feb. 2020. Date of publication 2 Mar. 2020; date of current version 17 Jan. 2022. (Corresponding author: Yue Li.)
Digital Object Identifier no. 10.1109/TDSC.2020.2975789

demonstrate the damage that can be caused by password resets through case studies on four categories of websites, in which we show that significant privacy and financial losses are possible to incur. Then, we exam security policies of eight major email providers. We conclude that a significant portion of leading email service providers fail to take deserved effort to provide user email account protection, leaving them vulnerable to a variety of attack vectors. Afterwards, we demonstrate the damage can be caused by password reset through case studies on four categories of websites, in which we show that significant privacy and financial loss are possible to incur. Then, we examine the security policy of eight major email providers. We conclude that a significant portion of leading email service providers fail to take deserved effort to provide user email account protection, leaving them vulnerable to a variety of attacking vectors. Furthermore, we have done a study on the evolution of the account recovery protocols by collecting additional datasets. We observe that in the past two years some websites (about 18 percent) have updated their account recovery methods but the majority remains unchanged.

Finally, we propose a password recovery protocol named Secure Email Account Recovery (SEAR) as a preliminary solution to address the single-point-of-failure problem of user email accounts. Specifically, the email provider adds an extra layer of protection, which can be in the form of an SMS authentication, etc., when a password reset email is intended to be opened. Thereby, the attacker cannot spread the attack from compromising an email account. We demonstrate our solution can be easily implemented under current network infrastructure with full backward-compatibility, and strengthens account security with all-rounded usability consideration (i.e., similar user experience, no need for handing out the phone number to all websites that one intended to protect, etc.). Through evaluation, we demonstrate that SEAR incurs very little resource overhead in terms of CPU and storage.

Overall, the major contributions of this work are summarized as follows.

1) We identify the de facto password recovery protocols in the wild by examining 239 traffic-heavy websites. In the measurement study, we build taxonomies on both websites and password recovery credentials, which enable us to explore the password recovery problem from different perspectives and dimensions. We also study the evolution of account recovery protocols and confirm that our findings have long-term validity.
2) We systematically investigate the email-based password recovery vulnerability that widely exists in today's web services. Our assessment reveals that the risk is high and could cause severe damage to users.
3) We propose SEAR as a preliminary solution that can be seamlessly integrated into modern email infrastructures in a fully backward-compatible manner. We implement SEAR based on popular open-source projects, and our evaluation shows that SEAR can be easily implemented and incurs negligible overhead.

This work is an extension of a preliminary conference paper [18], with focused improvement on the defense part of the account recovery study. Specifically, we elaborate the proposed SEAR protocal, from its motivation, to a more formal and more detailed protocol specification, as can be seen in Sections 7.1 and 7.3. In addition, we evaluate the performance of our prototype in terms of the CPU and storage overhead, as can be found in Section 8.2. Furthermore, we add discussions in Section 9 on the applicability of SEAR, and other security enhancement (usable 2FA) that is available.

The rest of the paper is organized as following. Section 2 clarifies related terms. Section 3 overviews password recovery protocols in modern websites. Section 4 evaluates the success rate of a password recovery attack. The extent of damage is presented in Section 5. Section 6 studies the evolution of account recovery protocols, and Section 7 elaborates SEAR, our defense mechanism. The implementation and evaluation of SEAR are detailed in Section 8. Section 9 discusses the applicability of SEAR and usable 2FA. Section 10 surveys closely related works, and finally, Section 11 concludes the paper.

2 Terminology and Definitions

With the development of new schemes and years of advances in multiple dimensions, the account recovery process cannot be easily elaborated. In order to help understand and organize the heterogeneous makeup of the account recovery process, we first perform a classification on account recovery credentials and websites.

2.1 Recovery Primitive, Method, and Protocol

Account recovery is essentially another authentication process, which needs one or multiple legitimacy validations. Each validation is usually done on the server side by matching a mutually agreed-upon piece of credential ε to the one supplied by the login attempter. While an ε can be represented by a series of symbols, we categorize ε into six types based on their sources, as listed below, and we call each type a *recovery primitive* (γ).

- *Email* (γ_{em}). Email primitive is the accessibility to a registered email. The validation process may be of various manners, such as sending a hyperlink to reset a password, sending a one-time code for inputting a password reset form, or even directly sending back the original password. Nevertheless, accessibility to the registered email is the only prerequisite.
- *Phone* (γ_{ph}). Similar to email, phone primitive demands accessibility to a phone that is associated with a preregistered phone number. The website may choose to call the phone number or send a text message.
- *Security question* (γ_{sq}). Security question is a kind of knowledge-based authentication, which allows a password reset if questions are answered correctly. Normally, the answers to security questions are intrinsic to users, and hence no extra memory burden is introduced. An example is, "What is your favorite food?"
- *Private information* (γ_{pi}). Private Information is also knowledge-based authentication in a personally identifiable and thus not massively predictable sense, the answer to which is relatively unique among

different users. Although users may or may not intrinsically remember it, they usually have access to the information from other channels. Examples of such information include a credit card number and Social Security Number.

- *Activity Information (γ_{ai})*. Activity information involves account activity traces. Some service providers believe that a user is expected to be able to recall some of the most basic activities of its account, such as the nickname/username, most login locations, and other users with whom they usually interact. It may even require assistance from acquaintances on the same website.
- *Recovery Token (γ_{rt})*. A recovery token is usually a non-memorizable piece of information that users possess. Examples are randomly generated tokens at registration or one-time codes generated by mobile applications (authenticators or website-designated apps).

In some cases, websites may ask for a combination of multiple γ for stronger authentication and provide multiple such combinations for users to choose from for increased flexibility. To set boundaries among these similar concepts, we define one way to recover a password as a *recovery method*. Fundamentally, a recovery method could consist of one or multiple recovery primitives, and all primitives should be supplied by a user correctly in order to recover its account. For example, on a website ω, one way to recover a user password may be $m_{\omega,1} = \{\gamma_{em}, \gamma_{sq}\}$, meaning that the password can be reset by whomever is in possession of the registered email and answers to the security questions. A recovery method is the most basic unit of a successful account recovery. Similarly, we define the set of all m that a website provides as the *recovery protocol* (p) of the website. For instance, for the website ω, its recovery protocol is $p_\omega = \{m_{\omega,1}, m_{\omega,2}, \ldots, m_{\omega,i}\}$, indicating that there are i recovery methods and that any recovery method can be used alone to successfully recover an account.

2.2 Website Classification

We categorize websites into several groups, helping us look deeper into how different websites handle account recovery in a finer granularity, as well as conduct the damage assessment. Grosse and Upadhyay [19] have done a user account classification based on the values of the accounts. However, their classification is user-oriented, which heavily relies on user-subjective perspective and activity. Namely, different users may have different types of accounts on the same website, depending on the user's purpose for using the websites. By contrast, we take a website-oriented approach by classifying websites based on their service nature. We define the following six website groups with some terminologies acquired from [19].

1) *Routine*. A routine website is one in which users passively receive information. Most of its users produce zero or little long-residing content. Examples of routine accounts are online newspapers, those used for online education, and gaming or music websites.
2) *Spokesman*. Spokesman website accounts usually represent a user's opinion or identity. Users rely on spokesman websites to deliver and exchange information with other real users. Examples of spokesman websites are online social networks, such as Facebook, Yelp, and LinkedIn.
3) *E-commerce*. E-commerce websites mainly involve trading. A business website could be an online retailer, such as Amazon and Ebay, or paid service providers, such as insurance companies. It is common to find addresses, shopping histories, phone numbers, and even payment information in user accounts on these websites.
4) *Financial*. A financial website usually concentrates on financial activities, such as deposits, withdrawals, and online transactions. Examples of financial websites are banking, brokerage, or wallet-type websites, such as Paypal.
5) *Tool*. A tool website does not usually produce a final product. Instead, it provides a tool or platform for helping build or shape the final product. Examples of tool websites are search engines, website builders, online graph drawers, and web traffic analyzers.
6) *Email*. Email websites provide online accounts that are associated with user email addresses, which can send and receive emails, such as Gmail or Outlook.

Nowadays, it is common for websites to have a heterogeneous service nature. It is sometimes hard to classify a website into a single type. For example, Google is a tool website since it offers a search engine. Meanwhile, it is also a spokesman website (Google+) and an email website (Gmail). As such, we sometimes classify a website as multiple types. While allowing such cases, we primarily categorize a website based on its main services and user recognition. For example, an online newspaper may have a review section under an article where users can express and discuss their opinions. However, most users may only browse the news without writing any comment. Thus, the online newspaper is categorized solely as a routine website, instead of a spokesman website.

3 Password Recovery in the Wild

We manually investigate the password recovery protocols adopted by the Alexa top 500 websites to help understand the protocol composition in modern websites. Since the top 500 websites are ranked by their global web traffic, each of them has a large number of users (or visitors), and thus reflects the de facto techniques adopted for password recovery.

3.1 Demographics

Within the 500 most traffic-heavy websites, we identify 245 websites in which we are able to create an account. Among them, 239 (97.5 percent) websites have enabled an account recovery protocol (p). Since we are only interested in recovery protocols, we consider our dataset to contain only the 239 websites thereafter. There are fewer protocols than websites due to multiple reasons. First, we count the same protocols that share the same database only once, such as all regional Google sites and subsidiaries of Google, such as Youtube. This type includes 99 websites. Google alone contributes 55 of them. Second, there are 40 websites that do not have login functionality. For example, some online newsletters do not

TABLE 1
Website Visitor Origin Distribution

Country	Total	Percentage	Examined	Percentage
U.S.	191	38.2%	135	56.5%
China	70	14.0%	35	14.6%
India	56	11.2%	26	10.9%
Japan	30	6.0%	11	4.6%
Russia	18	3.6%	7	2.9%

TABLE 2
Recovery Primitive Distribution

Primitive	Number	Percentage	Self-sufficient	Percentage
Email	232	97.07%	213	89.12%
Phone	46	19.25%	40	16.74%
Security Question	22	9.21%	11	4.60%
Private Information	7	2.93%	0	0.00%
Activity Information	12	5.02%	10	4.18%
Recovery Token	3	1.26%	3	1.26%

"self-sufficient" implies the recovery primitive is the sole ingredient in a recovery method (i.e., $|m| = 1$, for example, $m = \{\gamma_{em}\}$).

need user logins. In addition, some recorded sites are just advertisement network referrer links or content delivery networks in which not even an accessible homepage is available. Examples are adnetworkperformance.com and www.t.co. Third, we fail to examine 51 websites with less commonly used languages. It is challenging to recognize and input CAPTCHA in these languages, which is a required process in order to register an account. Finally, on the rest of the websites, a local phone number or membership is mandatory for registration. Examples include most online banking systems. These websites are not open to an outsider, and thus we are unable to access them.

The websites being successfully examined bear a similar distribution on visitor origins in the Alexa top 500 list (See Table 1 for the top 5 countries). Though only a limited number of websites are examined, these popular websites attract most web traffic. For instance, Google alone is reported to account for up to 40 percent of web traffic [20]. Therefore, we believe that our analysis is representative and can genuinely cover the mainstream of modern website account recovery protocols used by most online users.

Overall, our dataset contains 239 websites that enable account recovery, naturally including 239 password protocols. In these protocols, we identify 324 recovery methods. We identify 364 recovery primitives in these recovery methods. On average, a website has 1.36 recovery methods, and each method constitutes 1.12 recovery primitives. This implies that most of the websites provide only one recovery method, and recovery primitives in a recovery method are, to a large extent, homogeneous.

Note that nowadays, many websites have used Single Sign On (SSO) for logging in. SSO enables a user to use the account of an identity provider, such as Facebook, to log into other websites. In our dataset, 136 websites feature at least one SSO identity provider. The top three are Facebook (103 occurrences), Google (67 occurrences), and Twitter (35 occurrences). Regarding SSO, users cannot lose their accounts unless they lose access to their accounts of the identity providers. Thus, a user of these SSO websites should recover its account from the SSO identity provider, such as Facebook.

3.2 Primitive and Method Usage

To illustrate the major composition of recovery protocols, we examine the usage of recovery primitives and list the overall occurrence of each primitive in Table 2. As shown in the table, using email to recover a password is prevailing: 97.1 percent of websites include email (γ_{em}) in their recovery protocols. Furthermore, among 89.1 percent of websites, email itself is sufficient to recover a password (namely, at least one of their recovery methods contains the element of email primitive only). It is evident that most of the top websites delegate the security responsibility of account recovery to email service providers, instead of extending and relying on their own security infrastructures.

The second most popular method is using a mobile phone, which is seen in a notable portion (19.3 percent) of websites. Surprisingly, 4.6 percent of websites still rely exclusively on security questions to recover passwords, which are suggested against by many previous researches [21], [22], [23]. There is one website that even asks one security question to reset a password. Meanwhile, private information, activity information, and recovery tokens are much less used since they may involve more deployment costs and have privacy concerns. However, these primitives are commonly used in sensitive online services, such as financial institutes.

From Table 2, we can also easily infer that most recovery methods contain only one recovery primitive. In fact, recovery protocols in 95 percent of the websites we examined include at least a single-primitive recovery method. As recovery methods with multiple recovery primitives are rarely found, scattered, and hardly organizable, we focus more on unveiling the structure of a single-primitive recovery method. Following the annotations introduced in Section 2.2, we identify 127 routine websites, 82 e-commerce websites, 52 spokesman websites, 36 tool websites, 6 financial websites, and 11 email websites. Their single-primitive recovery methods are portrayed in Fig. 1. It is not surprising to see that different genres of websites use different single-primitive recovery methods to balance their own security and usability trade-off.

From the figure, we can also see that financial websites are quite different from the other five — only one website uses a single-primitive method for account recovery (private information). Financial websites are the only type of website that usually has multiple recovery primitives in a recovery method. The other five types of websites all heavily rely on email (more than 80 percent for email sites and more than 90 percent for the other four) for account recovery. Using a phone follows as the second most commonly seen account recovery method. Interestingly, email websites themselves heavily rely on a mobile phone to recover passwords and are significantly more prone to use account recovery primitives other than email services. One possible explanation is that the email is already the end point of an account recovery chain and that email service providers prefer not to lead their users to their competitors'

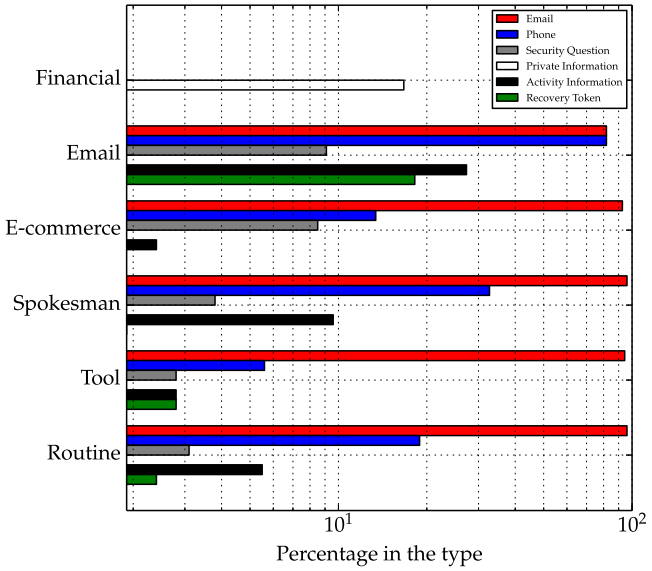


Fig. 1. Recovery methods – single-primitive.

email services. Thus, they attempt to offer other remedies, such as a mobile phone.

The usage of multiple-primitive recovery methods is not very common, given the fact that on average a recovery method consists of only 1.12 recovery primitives. Due to the sparsity, we list the total number of two-primitive and three-primitive methods used in each website category in Table 3. None of recovery methods we identified has more than three primitives. We can see that using email and security questions together is a common multiple-prmitive recovery method. Financial and e-commerce websites deploy such a multi-primitive recovery method more often than other websites, possibly due to their critical service nature. However, with equal importance, email services do not have any recovery method that has more than one primitive.

Overall, it can be inferred that most of the top websites delegate the security responsibility of account recovery to email service providers, instead of extending their own security infrastructures. There could be several reasons for this. First, it keeps high usability, as the registered email address becomes a centralized master key through which users can conveniently manage almost all of their online accounts. In other words, users incur almost no more cost when the number of online accounts increases. Second, email recovery mechanisms can be easily deployed at both sever and client sides. Third, the security obligation of account recovery is delegated to other online services, which significantly reduces the security responsibility of the website. Accordingly, the email of a user ends up as essentially another password manager and thus a potential single point of failure. The illegal access to an email account can pose a serious security threat on most websites, with the only exception being financial websites. As a result, an intruder can easily compromise user accounts by mounting an account recovery attack, especially nowadays when email accounts are at a massive loss [17].

TABLE 3
Recovery Methods - Multiple Primitives

Primitives	Routine	Tool	Spokesman	E-commerce	email	financial
$(\gamma_{em}, \gamma_{sq})$	3	1	2	5		1
$(\gamma_{em}, \gamma_{ai})$	2	2	1	1		
$(\gamma_{em}, \gamma_{ph})$						1
$(\gamma_{em}, \gamma_{pi})$				1		1
$(\gamma_{sq}, \gamma_{pi})$	1					1
$(\gamma_{sq}, \gamma_{ph})$						1
$(\gamma_{pi}, \gamma_{ph})$				1		1
$(\gamma_{em}, \gamma_{sq}, \gamma_{pi})$						1
$(\gamma_{em}, \gamma_{ai}, \gamma_{pi})$						1
$(\gamma_{em}, \gamma_{ph}, \gamma_{sq})$		1		1		
$(\gamma_{sq}, \gamma_{ai}, \gamma_{pi})$						1
$(\gamma_{ph}, \gamma_{ai}, \gamma_{pi})$						2

4 Attack Assessment

Since emails play a critical role in account recovery, it is necessary to evaluate the vulnerability that may be introduced by emails, especially when user email accounts are at risk.

Although we have observed that most websites rely on emails to recover user passwords, the assumption that possessing a password will compromise a user account may no longer hold under today's multi-dimensional authentication context, where a password may not always be the sole gatekeeper. In fact, an account recovery attack should be considered successful only if an intruder can actually log in to the target website and impersonate the victim user, which rules out cases where the intruder steals a password but fails to access the account due to a lack of other credentials. We first define the capabilities of an intruder and then discuss the possibility of a successful attack.

4.1 Threat Model

We assume that an intruder has access to the victim's primary email account and attempts to log in to a user account by exploiting the information included in the recovery email. Specifically, the intruder has no knowledge about the victim's personal information and makes no attempt to obtain the information that is believed discoverable or guessable yet hardly quantitatively assessable, such as user-chosen usernames (when it is neither the email address nor included in the recovery email) or security question answers. We also assume that the intruder does not make extra efforts to bypass additional classifier-based authentication schemes, such as IP address or OS/browser fingerprinting. Note that we aim to set a baseline for the success rate of an account recovery attack so that we keep our attack model simple and clean. In the real world, intruders may try to use more sophisticated tactics to break into even more user accounts [24].

4.2 Possibility to Break-in

As suggested by Table 2, 213 out of the 239 websites solely rely on emails to recover user passwords, making 89.12 percent of the examined websites potentially vulnerable to account recovery attacks. However, an attack may not be successful for two reasons: the lack of other credentials and additional classifier-based authentication. Thus, these factors should also be taken into consideration for estimating the success rate of mounting account recovery attacks on these 213 websites. We discuss the impact of each factor as follows.

4.2.1 *Lack of Credentials*

The first factor is the lack of other credentials, and a user's password is not the only credential needed to log in to a

system. We investigate the use of a username, as it is also a required piece of information for successful authentication. A website needs to know a username or an email address first to locate an account so that the corresponding recovery methods, such as security questions for the designated user account, can be retrieved. We identify that 80.3 percent (192 out of 239) of websites allow the use of email addresses as usernames, and 178 of them can use emails to recover their passwords. On the other hand, 23 websites that do not treat email address as a type of username (i.e., a username is freely selected by its user) provide email-based username recovery or directly send a username in the account recovery email, meaning that the username itself can be accessible from the email account. Thus, in total, 84.1 percent (201 out of 239) of the examined websites are potentially vulnerable to account recovery attacks. In other words, among the 213 websites that allow emails to recover passwords, 12 of them are immune to account recovery attacks because intruders cannot know usernames through emails. Another lack-of-credential scenario is the two-factor authentication (2FA) in which an intruder has no access to the other authentication factor. In this case, the login will also fail. We found that 35 websites feature 2FA options. However, the general adoption rate is still believed to be quite low. By analyzing more than 100,000 Google accounts, Petsas *et al.* [25] estimated that Google 2FA is adopted by no more than 6.4 percent of its users in 2015. It is also unlikely for other websites to have a much higher adoption rate than Google. Furthermore, 2FA is an option disabled by default in all of the websites we have identified. Therefore, a 2FA-available website should still be considered vulnerable to account recovery attacks since more than 90 percent of the users are not really protected.

4.2.2 Classification-Based Authentication

The other factor taken into account is classification-based authentication. Leveraging more on de-centralized credentials may incur significant usability hassles and thus repel users. Therefore, many websites start to use a classifier to automatically verify a legitimate login attempt to balance usability with security, where a correct password is not sufficient for login. The classifier aims to detect anomalous login behaviors by taking many signals into the classification decision, such as the IP address, cookies, and OS/browser fingerprints. Alaca *et al.* [24] identified and evaluated 29 fingerprinting mechanisms, and each of them may produce multiple signals. If a login attempt is classified as suspicious, the system is likely to trigger a standard 2FA. Authentication classification is reported by Google [26] to effectively reduce 99.7 percent of account compromises using more than 120 features. However, the classification is a black-box that is hard to comprehend, especially when the features are numerous. To determine whether a website has enabled a classifier, we adopt an attacker-centric approach, where we probe all 239 websites by using the Tor network and VPN,[1] which enables us to emulate an attacker. Specifically, we first train each website by manually logging in to the website on the same computer once per day for a period of one week. The computer has a fixed fingerprint and IP address. Then, we camouflage ourselves as a user in a different country with different operating systems and browsers (all cookies cleared) to log in to the same website three weeks after the training stage. Note that we have provided necessary information, which includes a backup email, phone, and security question, for the use of the 2FA to the website when the classifier has low confidence. Our methodology cannot guarantee 100 percent accuracy of the results since the classification systems of these websites are still unknown. However, we believe that our results are sufficiently close to the ground truth since a useful classifier should capture such obvious anomalies. Our results indicate that only 14 (5.9 percent) out of the 239 websites are using a classifier, as we are either required to complete a standard 2FA or blocked from logging in. Furthermore, 8 of the 14 websites rank top 30 in web traffic, and the others are mainly financial websites. Clearly, though useful, classification-based authentication has not been widely used, and thus account recovery vulnerabilities still remain, at least at the current stage.

1. The Tor network is known for having abnormal login issues in some websites, so we use both Tor and VPN to obtain most accurate information.

TABLE 4
Websites Vulnerable to Password Recovery Attacks

All	R_1	$R_1\&R_2$	$R_1\&R_2\&R_3$
239	213 (89.12%)	201 (84.1%)	194(81.1%)

R_1 : Allow emails as a password recovery method. R_2 : Username is directly obtainable. R_3 : No classifier is enabled.

After considering the above two factors, we are able to answer the question of how many websites are vulnerable under such an attack model. We concisely summarize the results in Table 4. As can be seen from the table, 81.1 percent of the websites we examined are vulnerable under our threat model. In addition, if an intruder is sophisticated and could emulate enough login signals to deceive the classifier, 84.1 percent of the websites would be vulnerable to account recovery attacks.

5 Damage Estimation and Email Security

As a large portion of websites are vulnerable to account recovery attacks when a registered email is compromised, we evaluate possible damages that could be caused and the security policies of major email providers, which are essential to throttle attacks on user email accounts.

5.1 Damage

The damage can be multi-fold. First, intruders are able to steal private information, such as home address and activity history of users. In fact, this is the main reason why an intruderr is interested in user passwords. Second, the intruder may also actively impersonate legitimate users to post information, such as sending spam messages on the user's behalf [26]. Third, they may cause financial loss by purchasing products and stealing credit card or bank information. Measuring the extent of the damage can be complex and error-prone since even the same type of websites could have very different user data and security policies.

TABLE 5
Damage Estimation

	Site	Sensitive Information	Activity	Financial
Routine	Netflix.com	Phone Number,Watch History, Credit Card Number, Credit Card Info		Subscribe/Update Service
	nytimes.com	Name, Location, Purchase History, Occupation, Income, Gender		
	weather.com	Name, Birthday, Gender, Phone Number, Home Address, Work Address		
	wikia.com	Location, Birthday, Name, Gender, Occupation, Post		
Tool	github.com	Company, Location Credit Card Number, Credit Card Info	Sabotage	Purchase Service/Data
	dropbox.com	All Files Stored, Access History	Sabotage	
	skype.com	Phone number, Birthday, Location, Connection's Phone Number, Birthday, Location, Gender		
	ebates.com	Name, Address, Shopping Histories	Spamming	
Spokesman	facebook.com	Name, Address, Birthday, Gender, Work, Education, Phone Number Contact's Information, Posts, Messages	Spamming, Sabotage	
	instagram.com	Name, Phone, Gender	Spamming, Sabotage	
	reddit.com	Private Message	Spamming, Sabotage	
	quora.com	Private Message	Spamming	
	livejournal.com	Birthday, Location, Private Message, School, Posts	Spamming	
E-commerce	amazon.com	Shopping History, Files on Cloud, Name, Address, Phone Number, Private Messages Reviews, Browsing History, Credit Card Number, Credit Card Information		Purchase Products/Services
	ebay.com	Name, Gender, Address, Buying History, Selling History, Private Message, Phone Number, Credit Card Number, Credit Card Info	Sabotage	Purchase Products*
	walmart.com	Name, Address, Phone, Buying History, Credit Card Number, Credit Card Info		
	gap.com	Name, Gender, Address, Phone Number, Shopping History Credit Card Number Credit Card Info		

Red color represents the information is fully obtainable while Blue color indicates the information is only partially obtainable.
** When purchasing a product at Ebay, the user can modify the shipping address without re-inputting payment information.*

We estimate the possible losses by examining typical websites from four major website groups, which are routine, tool, spokesman, and e-commerce. We do not examine the email group as Egelman *et al.* [27] have already done a thorough investigation on how much sensitive information resides in one's primary email account, reporting that a substantial amount of sensitive information can be found in the email archive, such as credit card numbers (16 percent) and SSN (20 percent). We also exclude the financial group due to the fact that all of the financial websites we examined in the Alexa top 500 websites are immune to the account recovery attack, as the email is insufficient to reset a password. In our examination, we select those websites with a single service type. In addition, we also try to select these websites that are likely used by normal users. A counter-example is a paid advertisement publisher, which has a high volume of web traffic, but few normal users would use it.

We show the damage assessment in Table 5. It is evident that all of the websites we examined, to various extents, expose user private information, such as phone numbers, birthdates, and addresses, to attackers. In addition to private information, an attacker is able to actively mount subsequent attacks, such as sabotaging, spamming. Sabotaging may not be as appealing to the attacker since it does not bring many benefits. However, using a real account for spamming or phishing attacks is a common practice among attackers due to the fact that real accounts are much more credible [26]. Furthermore, attackers may even make purchases in e-commerce websites with stored payment information. On the other hand, it is easier for the attacker to change the shipping address to receive the ordered package or intercept the delivery process. We observe that many e-commerce websites require a payment re-authentication, in terms of the credit card security code (Walmart and GAP), to post an order. Amazon requires to re-input the complete payment information if the shipping address is new. However, surprisingly, Ebay allows a user to change the shipping address freely without additional authentication, which makes financial losses largely possible if the account is compromised.

5.2 Assessing Email Security

Since email is pivotal to account recovery, the security of user accounts in a website is heavily dependent on the email security. A more secure email service can certainly help to thwart account recovery attacks in the first place.

To this end, we evaluate the security policies of all 11 major email service providers in our dataset, which span different geo-locations, including North America, Asia, and Europe. The fields examined involve several authentication policies, including minimum password length, minimum password composition (uppercase letters, lowercase letters, digits, and special characters), whether 2FA is provided, and whether a classifier is used to filter out abnormal login attempts. The list of providers we examined and results are shown in Table 6.

We also list the password policies of all six types of websites in Table 7, with respect to minimum password length and minimum types of characters required (on average). We can see that the minimum length of passwords in email websites is seven, which is only less than that of financial websites. However, email websites have the weakest composition complexity policy, since most of them do not require more than one type of character in a password, and users are more likely to create predictable passwords under such a policy.

We also notice that a significant portion of email providers include 2FA in their authentication systems. Compared to the overall rate of 2FA-enabled websites, email providers show a much higher security concerns and offer 2FA enhancement to secure user accounts. However, the

TABLE 6
Examining Major Email Providers

Provider	Region	Length	Composition	2FA	Classifier
Gmail.com	USA	8	1	✓	✓
Yahoo.com*	USA	7-10	4-1	✓	✓
outlook.com	USA	8	2	✓	✓
AOL.com	USA	8	1	✓	
QQ.com	China	6	1	✓	
163.com	China	6	1	✓	
sina.com.cn	China	6	1		✓**
china.com	China	6	1		
china.com.cn	China	6	1		
Rediff.com	India	6	1		
Yandex.com	Russia	8	1	✓	

** Minimum password length and composition can vary depending on each other. For example, a password of length 7 should have 4 types of characters to be accepted by Yahoo. However, a password of length 10 can have only 1 type of characters.*
*** The on/off of classifier is configurable, the default is off.*

TABLE 7
Password Policy in Average

	Routine	Spokesman	E-commerce	Financial	Tool	Email	Overall
Length	5.74	5.54	6.35	7.33	5.91	7.0	5.92
Composition	1.20	1.19	1.57	1.67	1.36	1.18	1.33

In Yahoo.com, we choose minimum length of eight and composition of two since this setting may fit more normal passwords.

number of users that actually use 2FA is likely to be small [25]. A more effective solution might be using a classifier to verify a legitimate authentication attempt. Although some of the classification signals can be easily spoofed [24], it is still difficult for an attacker to correctly spoof all signals considered by the classifier, especially when the adopted signals are unknown [3]. Unfortunately, only 4 out of the 11 email providers have integrated such a protection mechanism. One of them (sina.com.cn) requires a user to turn on the classifier, but most users probably do not enable it as the default setting is off. The other 7 email providers are much easier to be compromised by phishing attacks and password guessing/cracking attacks. Under such a condition, those accounts that are associated with a weak email account are vulnerable to account recovery attacks.

Generally speaking, a large portion of major email service providers fail to provide adequate security protection on user email accounts. It makes an account recovery attack more likely to happen, and thus jeopardizes the security of the online accounts that rely on emails for account recovery.

6 Account Recovery Protocol Evolution

Since our initial analysis of this work [18] was conducted in 2017, the account recovery implementations may have changed since then. Therefore, it is important to see how the recovery protocols have evolved during the past two years. It is also helpful for us to project the future validation of this study. To understand the trends, we conducted another round of data collection in April 2019. Similarly, we manually recorded the recovery protocols of the 239 analyzed websites. In general, we observed that 41 websites have changed their recovery protocols while the other 198 websites remain unchanged. Namely, most websites still rely on user emails for account recovery. While the overall picture is shifting slowly, we further investigated the detailed changes to understand the future trend.

Among the 41 websites, four websites are no longer accessible, and one website (www.reference.com) stops to provide login functionality. For the other 36 websites, 19 of them provide more recovery methods, and 6 of them have less recovery methods. As such, we can see that the websites offer more flexibility to users by providing more recovery methods. The rest 11 websites have a consistent number of methods with the methods themselves being changed. We do not further break down the websites into different genres, due to the fact that the small number of evolved websites will make the break-down sparse and unreliable. Instead, we treat them as a whole to uncover a more general trend.

We depict the evolution of recovery primitives in Fig. 2, from which we make the following observations.

- There is an increasing number of websites allowing the single email primitive as a recovery method. Though more websites start to rely on emails for user account recovery, the usage of emails in a multi-primitive method has decreased. This is mainly caused by many websites that have deprecated the {email, username} (username is considered an activity information) recovery method. This also explains the reduction in activity information primitive.
- More websites have added the phone as one way to recover user accounts. This is as expected since mobile phones have become increasingly important in people's daily life, which also makes it weight more in user identification and authentication.
- It is encouraging to see that some websites have abandoned using security questions as the sole recovery primitive in one of their recovery methods due to its vulnerable nature. Instead, security questions have been more included in multiple-primitive recovery methods. We believe that this practice will enhance security without much usability degradation.
- We receive mixed signals from private information, activity information, and tokens, which could be increasing and decreasing simultaneously. At a closer look, we find that these changes are more related to a website service's nature. Websites that have more social contexts have incentives to include more activity information while others stop using it for better usability (such as the fore-mentioned {email, username} recovery method).

In summary, we have observed a slow evolution on the account recovery protocols in the past two years. The big picture mostly remains unchanged. However, improvements are

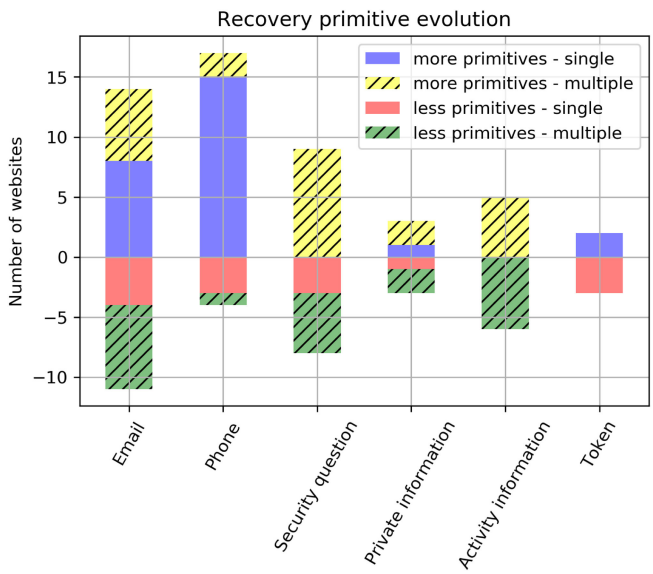


Fig. 2. Change of recovery methods.

indeed happening in some websites, such as the declined use of security questions. Aside from the trend we have learned from the new dataset, the observation that most websites solely rely on accessibility of an email for user account recovery still holds true. In fact, the number of such websites even increases. Thus, we confirm that our observation on account recovery protocols is not likely to drastically change. As such, our analysis results should be valid for a reasonably long-term period.

7 Securing Email-Based Account Recovery

Being aware of password recovery attacks, we propose a lightweight Secure Email Account Recovery (SEAR) protocol that can be integrated seamlessly into current network infrastructures. The core of SEAR is just adding a special protection layer on account recovery emails. Besides security as our primary concern, we take usability, deployability, and compatibility into our design consideration. By carefully gauging these properties, we ensure that SEAR can be easily put into practice without degrading regular user experience on email.

7.1 Motivation

Email-based password recovery gains dominating deployment mainly because of the usability concern. The usability advantage applies to both a website and its users. On the website side, it delegates the security responsibility to the email provider, which frees itself from technical or financial constraints, such as the cost of maintaining its own security infrastructure or the cost of contracting with telecommunication corporations to use SMS authentication. On the user side, they are unwilling to share a second factor credential (for recovery) to a web service they less trust, or they are reluctant to remember such a second factor. Thus, they are also happy to allow an email account to manage all other accounts. Email providers are even more attracted to this scheme, since users are forced to have an email account first in order to register another account.

It is not an uncommon scenario, in which (1) the email provider fails to adequately protect user accounts and (2) users themselves do not protect their accounts using a more secure approach, such as 2FA, due to the lower usability. It makes a user email account vulnerable to compromise; once compromised, attacks can be easily mounted to other user accounts by recovering their passwords. SEAR aims to cut this attack chain and to prevent the attacks from recovering other account passwords after an email account is compromised. SEAR requires very little effort from a website and its users, while the email provider is still the gatekeeper for password recovery. In short, SEAR requires a second factor authentication only in the rare case when a password recovery email needs to be accessed. In the meantime, the normal emailing experience remains unchanged.

7.2 SEAR Overview

SEAR relies on two premises. One is that the email provider is able to verify the user identity in two different ways, which is necessary for stronger protection on recovery emails. We assume that the users have provided the email provider two sets of credentials. It is mostly true since the email provider has its own recovery protocol that demands another set of authentication credentials other than username/password combination, such as backup email, phone number, etc. In addition, users are more willing to share credentials with service providers that they trust, which apparently include their email service provider.

The other premise of SEAR is that both account providers and email providers agree on a protocol for labeling an email as a password recovery email such that the email providers know which emails to protect. Meanwhile, SEAR should also ensure only legitimate service providers are able to label password recovery emails. Otherwise, a user may empirically consider an email protected by another authentication process as a legitimate recovery email and thus easily fall into phishing attacks. Denial of Service (DoS) attacks are also possible if such constraints are not enforced.

For the aforementioned protocol, we propose to add a header in an email to indicate that the email is for password recovery purpose. This method is transparent to existing email infrastructures since Simple Mail Transfer Protocol (SMTP) allows users to customize headers. The workflow of SEAR is briefly described as follows. First, the account provider will add a header "tag:value" pair (we choose "Recover:1") in the email. Upon receiving an email with the recover header, the email provider will request an extra authentication to protect it.

In our design, we consider prevention of possible SEAR-specific phishing vulnerabilities by enforcing the DomainKeys Identified Mail (DKIM) [28] protocol. The DKIM protocol uses public-key cryptography to ensure that the sender of an email is legitimate. DKIM has gained mass deployment and is enabled in over 80 percent email traffic of Gmail as reported by Google [29]. We also address the compatibility issues with existing Mail User Agents (MUAs) by introducing an "intermediate token". The intermediate token is used to be sent to the MUAs as the key to the second factor authentication process, because the MUA itself does not provide protection on emails that are fetched from the servers. Furthermore, SEAR is fully backward compatible to modern web services. Either entity that does not follow the specification of SEAR simply falls back to the normal email experience.

7.3 Protocol Specification

There are three parties involved in the email-based account recovery process. They are the user who wishes to recover the password, the account provider that manages the to-be-recovered account, and the email provider that provides the email service for the user.

For easy comparison, we first sketch a conventional email-based recovery procedure in Fig. 3a. First, the user sends a password reset request R_E that specifies her email address E to the account provider (step 1). This step is usually done by clicking a "forget my password" button on the website and inputting E in the following page. If the account provider recognizes the account that associates with E, it prepares a recovery email (e) and sends it to the address of E, which is managed by the email provider (step 2). Essentially $e = (T, H)$, where $H = \{h_1, h_2, h_3, \ldots\}$ is a set of headers that describes attributes of the email such as "From" and "Subject", and T is a token (usually in

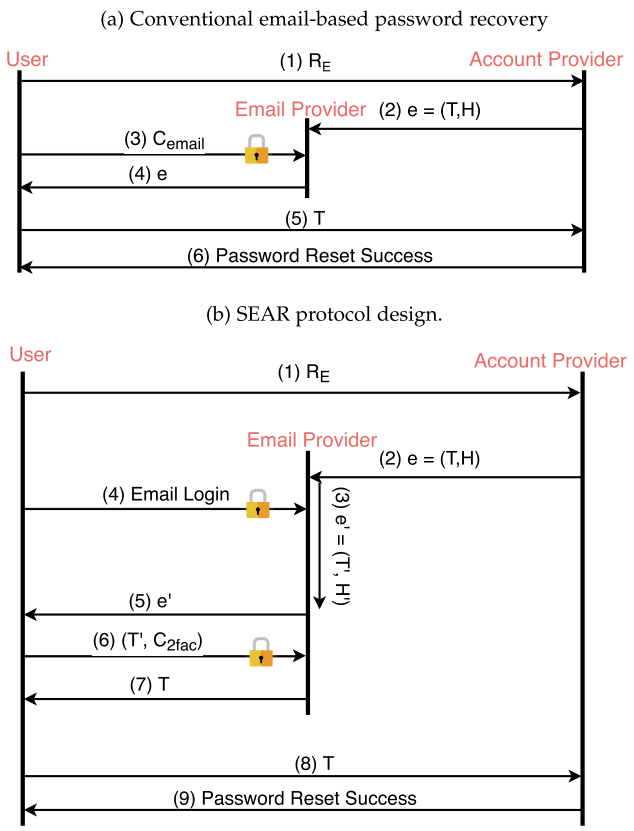


Fig. 3. Password recovery.

the form of a URL or a one-time passcode) to be submitted to the account provider to prove that the user has the access to the recovery email. Then, the user leverages her email account credentials C_{email} to log in the email service to open the recovery email (steps 3 and 4), and uses T to reset the password on the account provider (steps 5 and 6). It can be easily seen that whoever is able to sign in the user's email account can also reset passwords of other accounts of the user.

We now present the design of SEAR as shown in Fig. 3b, which mainly consists of two protocols, one for generating a recovery email and the other for verifying and protecting the recovery email.

7.3.1 Recovery Email Generation

Similar to the conventional method, the user signals a password recovery request R_E to the account provider (step 1). After receiving $R(E)$, P_a generates the recovery token T, which together with the headers H', are loaded in the body of a recovery email (e) and sent to the address of E (step 2). Note that we add a special header h_{re} to indicate it is a recovery email, and thus $H' = H \cup \{h_{re}\}$. When DKIM is enforced, a special header "DKIM-Signature" h_{DKIM} is also included in the headers H'. The "DKIM-Signature" header consists of the actual certificate and other necessary parameters for signing/verifying the signature, which prevents the email from being spoofed.

7.3.2 Verification and Protection

After receiving the email from the account provider, the email provider first conducts DKIM verification. If the email content hash values match (i.e., the email is legit) and the h_re is present, the email provider has received a legit password recovery email that should be protected with extra effort.

One main challenge in the email delivery process is to maintain compatibility to existing email retrieval protocols on the Mail User Agent (MUA), such as Thunderbird, outlook, and Yahoo! Mail, etc. MUAs use IMAP or POP3 protocols to fetch emails from the Mail Delivery Agent (MDA) of email providers. However, these two protocols do not add access control on specific emails, which makes protecting e with a second factor infeasible on existing MUAs. In other words, when an MUA fetches e to local, there is no way to further protect T inside e. To address this issue without changing the existing protocols, we introduce an intermediate token T' in the form of a URL. Specifically, the email provider replaces T with T', and store T in its servers (step 3). Consequently, the user MUA fetches a modified email $e' = (H', T')$ (steps 4 and 5). The user presents T' to the email provider, indicating the intention to access T. At this moment, the email provider enforces the user to input credentials C_{2fac} for a second factor authentication (step 6). Since T' is associated with the user, the second authentication factor registered by the user can be pulled out accordingly. A successful authentication grants T (step 7). At this point, the user is in possession of T, which can be directly used for the account provider to reset the password as in the conventional way (steps 8 and 9).

For web-based MUAs (i.e., email websites such as www.gmail.com) being fully controlled by email providers, they have more freedom to choose different implementations other than the one we proposed.

7.4 Security Analysis Using AVISPA

SEAR is a security protocol that adds extra protection to account recovery emails. A security analysis is important to understand the effectiveness of SEAR and potential vulnerabilities that might be incurred if SEAR is adopted. We conducted a security analysis over SEAR protocol using Automated Validation of Internet Security Protocols and Applications (AVISPA) [30].

7.4.1 Introduction to AVISPA and SEAR Implementation Model

AVISPA is a tool used for automated validation of security protocols and it attempts to find out vulnerabilities against the protocols being tested in verification. In the AVISPA tool, protocols are specified using the High Level Protocol Specification Language (HLPSL). Compared with "Alice-Bob" (A-B) notation, HLPSL is defined role by role rather than message by message, which contains more information and thus it is far more precise than usual A-B notation. Fig. 4 illustrates the specification of the SEAR protocol using HLPSL. The working procedure of HLPSL is that we specify the state of each entity being involved, and how it receives and sends data to one another. Transitions among states will happen based on the interactions between entities. The *goal* specifies the aim-to-protect credential. If at the end of the session, no credential is leaked, then the protocol is considered secure.

According to A-B convention, A, B and C represent a user, an email provider, and an account provider, respectively. Since AVISPA does not differentiate asymmetric and

```
role user(A,B,C:agent,Cemail,C2fac:symmetric_key,SND_AC,SND_AB,RCV_AB:channel(dy))
played_by A
def=
        local
                State:nat,Nab,Nba,T:text
        init
                State := 0
        transition
                1. State=0 /\ RCV_AB(start) =|> State':=1 /\ SND_AC(A)
                2. State=1 /\ RCV_AB({C.Nba'}_Cemail) =|> State':=2 /\ Nab':=new()
/\ SND_AB({Nba'.Nab'}_Cemail) /\ request(A,B,nab,Nab') /\ witness(A,B,nba,Nba')
                3. State=2 /\ RCV_AB({Nab.T'}_C2fac) =|> State':=3
/\ request(A,B,abt,T')

end role

role emailprovider
(A,B,C:agent,Cemail,C2fac,Cdkim:symmetric_key,SND_BA,SND_BC,RCV_BC,RCV_BA:channel(dy))
played_by B
def=
        local
                State:nat,Nab,Nba,Nbc,Ncb,T:text
        init
                State := 0
        transition
                1. State=0 /\ RCV_BC({B.C.Ncb'}_Cdkim) =|> State':=1 /\ Nbc':=new()
/\ SND_BC({B.C.Ncb'.Nbc'}_Cdkim) /\ request(B,C,nbc,Nbc') /\ witness(B,C.ncb,Ncb')
                2. State=1 /\ RCV_BC({A.Nbc.T'}_Cdkim) =|> State':=2
/\ Nba':=new() /\ SND_BA({C.Nba'}_Cemail) /\ request(B,A,nba,Nba')
                3. State=2 /\ RCV_BA({Nba.Nab'}_Cemail) =|> State':=3
/\ SND_BA({Nab'.T}_C2fac) /\ witness(B,A,abt,T) /\ witness(B,A,nab,Nab')

end role

role accountprovider(A,B,C:agent,Cdkim:symmetric_key,SND_CB,RCV_CA,RCV_CB:channel(dy))
played_by C
def=
        local
                State:nat,Ncb,Nbc,T:text
        init
                State := 0
        transition
                1. State=0 /\ RCV_CA(A) =|> State':=1 /\ Ncb':=new()
/\ SND_CB({B.C.Ncb'}_Cdkim) /\ request(C,B,ncb.Ncb')
                2. State=1 /\ RCV_CB({B.C.Ncb.Nbc'}_Cdkim) =|> State':=2 /\ T':=new()
/\ SND_CB({A.Nbc'.T'}_Cdkim) /\ request(C,B,nbc,Nbc') /\ secret(T',t,{A,B,C})

end role

role session (A,B,C:agent,Cemail,C2fac,Cdkim:symmetric_key)
def=
        local
                SAC, SAB, RAB, SCB, RCA, RCB, SBA, SBC, RBA, RBC: channel(dy)
        composition
                user(A,B,C,Cemail,C2fac,SAC,SAB,RAB)
          /\    accountprovider(A,B,C,Cdkim,SCB,RCA,RCB)
          /\    emailprovider(A,B,C,Cemail,C2fac,Cdkim,SBA,SBC,RBC,RBA)
end role
role environment()
def=
        const
                t,abt,nab,nba,nbc,ncb:protocol_id,
                a,b,c:agent,
                cemail,c2fac,cdkim,cemail_i,c2fac_i:symmetric_key

        intruder_knowledge ={a,b,c,cemail,cemail_i,c2fac_i}
        composition
                session(a,b,c,cemail,c2fac,cdkim)
          /\    session(i,b,c,cemail_i,c2fac_i,cdkim)
end role

goal
                secrecy_of t
                authentication_on abt,nab,nba,nbc,ncb
end goal

environment()
```

Fig. 4. HLPSL description of SEAR.

symmetric encryption, we model SEAR using symmetric encryption with the encryption keys C_{email}, C_{2fac}, and C_{DKIM}. T is a nonce generated to represent the password recovery link, while N_{ab} and N_{ba} are nonces generated by A and B, used for verifying their identities. N_{bc} and N_{cb} are nonces generated by B and C, used for verifying their identities.

The following should be noted in our model verification.

- We only considered the attack model between the user and the email provider. The protection between the account provider and the email provider is enforced by the DKIM signature.
- We assume the intruder can be a legitimate user, who has its own passwords C_{email_i} and C_{2fac_i}, and has compromised one of user' passwords C_{email}.

7.4.2 Effectiveness Against Password Recovery Attacks

We run the HLPSL description of SEAR protocol on AVISPA. If a protocol is proven to be insecure, AVISPA would generate a bug report that specifies at what stage a credential could be leaked; otherwise, AVISPA would simply output a success message to indicate that the protocol is secure. The security of SEAR is proven by AVISPA as no bug report is generated. The generated link cannot be compromised given the fact the attacker has no access to C_{2fac}, which is the core element of SEAR. Meanwhile, AVISPA also generates simulation results, showing the communication messages between entities (i.e., the information flow in the protocol). The simulation results generated by AVISPA are shown in Fig. 5, which match the information flow of SEAR.

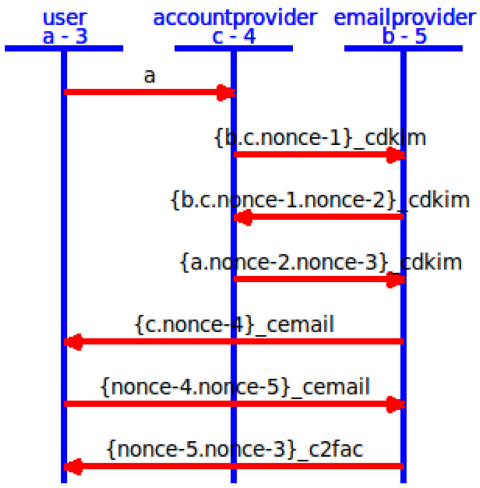


Fig. 5. Protocol simulation.

7.4.3 Fallback Mechanism

SEAR requires both the account provider and the email provider to follow the specifications in order to provide additional security. Otherwise, SEAR would not work well in the following three scenarios. First, if only the account provider follows SEAR and includes the recovery header in an email, but the email provider simply ignores the header and handles the email as a normal email, then SEAR does not work. Second, if only the email provider follows SEAR, the email provider may never receive any email with the recovery header from the account provider, since all emails from the account provider will be handled as normal emails. Third, when neither the account provider nor the email provider follows SEAR, it falls back to the conventional email-based account recovery. This fallback mechanism of SEAR guarantees its backward compatibility, producing no impact upon existing functionalities at both the account provider side and the email provider side. It enables SEAR to be incrementally deployable and be easily integrated into existing network infrastructures.

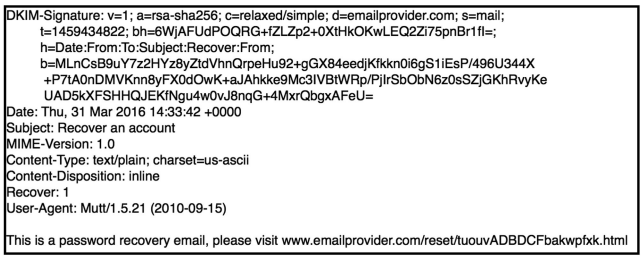
8 Implementation and Evaluation

We implement SEAR and build a simple email network to demonstrate its simplicity and compatibility. We also evaluate the system in terms of CPU and storage overhead in our implementation.

8.1 Implementation

We use two Amazon EC2 Ubuntu server [31] instances to act as the account provider and email provider correspondingly.

(a) Actual recovery email (e)



(b) Email to user(e)

Fig. 6. Password recovery - examples.

Both servers run Postfix [32], a widely adopted open-source MTA, for delivering and receiving emails.

At the account provider side, we use Mutt, a text-based email agent to generate normal and recovery emails. T inside a recovery email is in the form of a URL link. The emails sent out using Postfix on the account provider is protected by openDKIM [33], an open source DKIM library in C language. We configure openDKIM to sign all important headers including the "recover" header, as well as T as in the email body. Fig. 6a shows such a recovery email generated by Mutt with the DKIM-signature included.

On the email provider side, we modify Postfix source code to feature recovery header checking and subsequent actions. Specifically, we modify the "cleanup" daemon of Postfix, which is invoked by the master daemon. "cleanup" does processing before an incoming email is put into a queue and notifying the "queue manger" server the arrival of new emails. The "queue manager" afterward sends the email to a delivery agent to deliver the email to users, through IMAP or POP3 protocol if the user uses an MUA to check emails. In our implementation, we modify the "cleanup" procedure to check if the "Recover" header is present. If present, the procedure intercepts the email content and dumps a (user: T) pair in a local file. Meanwhile, "cleanup" procedure removes the T from the email and inserts an URL link (the T') instead. The URL link is associated with the email user. Then the modified email is put in the delivery queue. This procedure is described in Algorithm 1. We build a watcher server to monitor local file dump leveraging "inotify" API in Linux kernel.

The actual email (e') sent to the user is shown in Fig. 6b. It instructs the user to open a page that is guarded by another authentication factor pre-registered to the email provider by the user. Namely, the user is enforced to complete the extra authentication in order to view the content of the original password recovery email (e). As in our current prototype, the second factor authentication functionality is not implemented since it is a mature technique which does not need proof of applicability. After authentication, the user can directly reset password through an URL in this protected page (as in the conventional way).

By contrast, incoming emails without a "recover" header (i.e., not a password recovery email) can be directly appended to the incoming queue on the email provider server, which is exactly the same as original Postfix mechanism. In this case, the user can view their emails as they normally do.

Algorithm 1. Incoming Email Processing

1: **procedure** PROCESS ($Email$)
2: **if** ($Recover$ in $Email.header$) & ($Validate(Email.\ header.\ Recover)$) **then**
3: $dump(Email.body, user)$
4: $link \leftarrow generateURL(user)$
5: $Email.body \leftarrow link$
6: **end if**
7: $DeliverQueue.append(Email)$
8: **end procedure**

8.2 Evaluation

We evaluate the CPU and storage overhead of SEAR. We emphasize that the recovery email counts only a very small portion of all emails. The overall overhead should be multiplied by a factor β, which represents the percentage of password recovery emails in all emails. There is currently no large-scale data available showing that how many emails are involving password recovery. An analysis of our own 7 everyday email accounts shows that β may lie between 0.3 and 0.5 percent in average. Due to the small sample size, the estimation may not be accurate; however, we believe it suffices to get a sense that the percentage of password recovery email is very small.

8.2.1 CPU

We send 10,000 recovery emails to a SEAR-enabled Postfix server and 10,000 normal emails to a unmodified Postfix Server, and measure the CPU time used in the "cleanup" process. In average, a recovery email ($\mu = 797.2, \tilde{x} = 813, \sigma = 430.8$) consumes 8.1 percent more CPU running time than a normal email ($\mu = 737.0, \tilde{x} = 816, \sigma = 217.0$). Thus, the overall system overhead should be $76\beta\mu s$, which is roughly estimated between $0.23\mu s$ to $0.38\mu s$. In addition, the overhead can be further reduced due to the fact that dumping T to local storage is inevitable regardless, which should not be counted as extra work done. However, the CPU overhead is already considered minor by this rough estimation.

8.2.2 Storage

As H and T are stored anyway The storage overhead lies on hosting T'. T' is only a token pointing to T, which can be made as small as tens of bytes. In our experiment, we choose a moderate 32 bytes. To estimate the sizes of H and B, we meas Wiure the size of emails from Enron Corp, which is a public email dataset containing 517,407 emails from around 150 users working in Enron Corp [34]. The average email size is 2746.8 bytes ($\sigma = 8418.6$) in the dataset. As such, a recovery email takes 1.16 percent (32/2746.8) storage overhead. Again, this marginal overhead multiplies β implies the real overall storage overhead, which lies between 3.5×10^{-5} to 5.8×10^{-5}.

The results are based on our simple implementation on Postfix, which may be different if a different system is used, especially for very large email providers such as

Gmail or Yahoo, etc. These websites are likely to have used a plethora of optimization implemented on their distributed systems, which may cause the overhead to be different from our analysis. However, through qualitative and quantitative analysis, we show the resource efficiency and simplicity of SEAR. It suggests that SEAR is unlikely to be greedy on any system.

9 Discussion

9.1 Extended Framework

Emails may contain insensitive, semi sensitive, or sensitive information [27]. Therefore, email protection at different level is naturally needed. SEAR works as a deployable and compatible two-layered security solution under today's email infrastructures. In the current design, SEAR treats password recovery emails and normal emails separately, such a design can be extended to be more generic. For example, SEAR can be directly applied to emails including sensitive information (such as credit card number and SSN). Furthermore, the protection level can also vary based on the demands. For example, opening a sensitive email may require a 2FA while opening a semi-sensitive email may be directly granted by a classifier. As such, an extended framework of SEAR is promising in usable security enhancement under differentiated email sensitivity. Building such a more generic framework will be our future work.

9.2 Economical Validation

The motivation for email providers to adopt SEAR is a vital factor towards its popularization. We believe that SEAR will benefit both the end users and websites. On one hand, users are prone to choose more secure websites to protect their online accounts. Thus, those websites that have enabled SEAR are able to attract more users. On the other hand, the websites that have not adopted SEAR are motivated to integrate it to keep up with their rivals. In addition, SEAR does not introduce any compatibility issue, which eliminates the concern of malfunctioning in legacy systems. As a result, any websites, including both email providers and normal accounts, can deploy SEAR on their systems without sacrificing the functionality. It will motivate more websites to adopt SEAR due to its low cost.

9.3 Making 2FA More Usable

The non-negligible usability degradation hinders 2FA adoption. However, we have observed that some websites featured with 2FA also take a step towards making 2FA more usable. More specifically, the websites may remember the device that is used for logging-ins, which is usually through IP matching or cookies. On these remembered devices, a second factor authentication may not be triggered. Alternatively, the possession of the device becomes a factor. A user still only needs the password to log in the websites. However, to log in on a new device, a second factor should be correctly input for a successful authentication. To some extent, it is similar to a classifier as it considers a new device as "abnormal" login. In comparison to a classifier, it is not strong and accurate since many other useful signals are ignored. However, it is easier for the users to understand and manage. We believe this could be a useful future trend in multi-factor authentication as it has both competitive security and usability.

10 Related Work

10.1 Password Study

Ever since first deployed in the 1960s, password authentication has been extensively studied. People have reached a consensus that passwords are far behind the security level demanded due to human memorability limitation. Nearly 4 decades ago, Morris and Thompson [1] first indicate that passwords are predictable and thus vulnerable to dictionary attacks. Modeling and guessing passwords then become a mainstream weakness illustrations on passwords [4], [5], [35], [36], [37]. These vulnerability auditings also help shape modern password policies. Guiding users to create secure but memorable passwords attracts numerous potential solutions, such as mnemonic passwords [38], [39], user-replaceable passwords [40], and password managers [41].

10.2 Password Recovery

Providing no rescue when users forget their passwords is troublesome, especially when users own an increasing number of online accounts. Garfinkel [21] proposed Email-based Identification and Authentication (EBIA), which authenticates based on the ability to access a certain email address. Though not being able to universally replace password authentication, EBIA has been a primary way to password recovery. Similarly, receiving calls or SMS on cell phones is another de facto recovery scheme.

One previously popular recovery scheme is through security question [42]. However, it has been shown that secret questions are weak [22], [23] since the entropy is low and thus can be easily cracked through guessing or social engineering. Based on a large dataset from Google, Bonneau *et al.* [23] also show that secret questions have low recall rate and easily constructible distribution. They also found that users try to supply fake answers to make the questions harder, which however, yields the opposite outcome. Bonneau and Preibusch [43] have briefly discussed password recovery as a side ingredient in their study of password implementations. However, their work only takes a quick look at email and personal knowledge based password recovery and lacks systematic measurements and insights.

10.3 Alternative Schemes

Facing the dilemma that text-based passwords are not likely to be both secure and usable, numerous alternative authentication schemes are proposed. For example, one time password (OTP) [44], graphical [45], Biometric [46], [47], and behavior authentication [48]. However, none of these alternative schemes could bring all benefits with text-based passwords, making completely replacing text-based passwords unlikely in the near future [2]. However, some of these schemes can be integrated into text-based password framework serving as a second factor of authentication, which significantly enhances the security level. Now multi-factor authentication has played an important role towards safeguarding authentication legitimacy.

10.4 Multi-Factor Authentication

Multi-Factor Authentication enhances authentication security by requiring two or even more factors. Although there are cases that many factors are considered such as Bank of

America password recovery [49], two-factor authentication (2FA) is generally believed to achieve satisfying security level. Despite password as one factor, the other factor ranges from additional knowledge [50], [51], biometrics [52], to hardware tokens [53]. However, enabling 2FA sacrifices usability since it takes considerably more time and effort to complete. In order to mitigate such authentication fatigue, people try to make the second factor transparent by implicit data exchanging between the computer and the cellphone [54], or matching ambient sound recorded from the two devices [55]. Making 2FA transparent introduces deployability difficulty as it demands modifications on the server side. Instead, the authentication that stimulates more operational modes, such as progressive authentication [56], multi-level authentication, and opportunistic two-factor authentication [19], gains popularity. Although almost all of the features can be forged, it is hard to successfully forge all of them in practice.

11 Conclusion

In this paper, we first conduct a measurement study to characterize the password recovery activities on the Internet. Through extensively analysis on the password recovery protocols of major web service providers, we observe that nowadays a significant portion (81.1 percent) of website accounts can be easily compromised through email-based password recovery. However, many email providers fail to realize the threats and have not yet taken serious actions to protect their users, leaving the possibility of losing a large amount of personal information or even causing financial losses. To mitigate this single point of failure problem in email accounts, we introduce SEAR, a protocol established between email and account providers, to provide extra protection to password recovery emails. To demonstrate the deployability and compatibility features of SEAR, we build it on a popular open source email server. Our evaluation shows that SEAR introduces negligible CPU and memory overheads.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments, which help to improve the quality of this article. This work was supported by the U.S. Army Research Office under Grants W911NF-13-1-0421, W911NF-17-1-0447, and W911NF-19-1-0049, by the U.S. Office of Naval Research under Grants N00014-15-1-2007, N00014-16-1-3214, and N00014-18-2893, and by the U.S. National Science Foundation under Grants CNS-1618117 and CNS-1822094.

References

[1] R. Morris and K. Thompson, "Password security: A case history," *Commun. ACM*, vol. 22, pp. 594–597, 1979.
[2] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Proc. IEEE Symp. Secur. Privacy*, 2012, pp. 553–567.
[3] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "Passwords and the evolution of imperfect authentication," *Commun. ACM*, vol. 58, pp. 78–87, 2015.
[4] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *Proc. 12nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2005, pp. 364–372.
[5] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in *Proc. 30th IEEE Symp. Secur. Privacy*, 2009, pp. 391–405.
[6] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proc. 17th ACM SIGSAC Conf. Comput. Commun. Secur.*, 2010, pp. 162–175.
[7] P. G. Kelley *et al.*, "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms," in *Proc. IEEE Symp. Secur. Privacy*, 2012, pp. 523–537.
[8] D. Wang, P. Wang, D. He, and Y. Tian, "Birthday, name and bifacial-security: Understanding passwords of chinese web users," in *Proc. 28th USENIX Secur. Symp.*, 2019, pp. 1537–1555.
[9] S. Pearman *et al.*, "Let's go in for a closer look: Observing passwords in their natural habitat," in *Proc. 24th ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 295–310.
[10] A. Beautement, M. A. Sasse, and M. Wonham, "The compliance budget: Managing security behaviour in organisations," in *Proc. New Secur. Paradigms Workshop*, 2008, pp. 47–58.
[11] R. Shay *et al.*, "Designing password policies for strength and usability," *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 4, pp. 1–34, 2016.
[12] X. D. C. de Carnavalet and M. Mannan, "From very weak to very strong: Analyzing password-strength meters," in *Proc. 21th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2014.
[13] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley, "Does my password go up to eleven?: The impact of password meters on password selection," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2013, pp. 2379–2388.
[14] D. Wang, D. He, H. Cheng, and P. Wang, "fuzzyPSM: A new password strength meter using fuzzy probabilistic context-free grammars," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2016, pp. 595–606.
[15] D. L. Wheeler, "zxcvbn: Low-budget password strength estimation," in *Proc. 25th USENIX Secur. Symp.*, 2016, pp. 157–173.
[16] Z. Li, W. He, D. Akhawe, and D. Song, "The emperor's new password manager: Security analysis of web-based password managers," in *Proc. 23rd USENIX Conf. Secur. Symp.*, 2014, pp. 465–479.
[17] Hold security recovers 272 million stolen credentials from a collector. [Online]. Available: http://holdsecurity.com/news/the_collector_breach/
[18] Y. Li, H. Wang, and K. Sun, "Email as a master key: Analyzing account recovery in the wild," in *Proc. 37th Annu. Int. Conf. Comput. Commun.*, 2018, pp. 1646–1654.
[19] E. Grosse and M. Upadhyay, "Authentication at scale," *IEEE Security Privacy Mag.*, vol. 11, no. 1, pp. 15–22, Jan./Feb. 2013.
[20] Googles downtime caused a 40% drop in global traffic. [Online]. Available: https://engineering.gosquared.com/googles-downtime-40-drop-in-traffic
[21] S. L. Garfinkel, "Email-based identification and authentication: An alternative to PKI?" *IEEE Security Privacy Mag.*, vol. 1, no. 6, pp. 20–26, Nov./Dec. 2003.
[22] S. Schechter, A. J. Bernheim Brush, and S. Egelman, "It's no secret. Measuring the security and reliability of authentication via "secret" questions," in *Proc. IEEE Symp. Secur. Privacy*, 2009, pp. 375–390.
[23] 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 *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 141–150.
[24] F. Alaca and P. van Oorschot, "Device fingerprinting for augmenting web authentication: Classification and analysis of methods," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl.*, 2016, pp. 289–301.
[25] T. Petsas, G. Tsirantonakis, E. Athanasopoulos, and S. Ioannidis, "Two-factor authentication: Is the world ready? Quantifying 2FA adoption," in *Proc. 8th Eur. Workshop Syst. Secur.*, 2015, Art. no. 4.
[26] An update on our war against account hijackers. [Online]. Available: https://googleblog.blogspot.com/2013/02/an-update-on-our-war-against-account.html
[27] S. Egelman, S. Jain, R. S. Portnoff, K. Liao, S. Consolvo, and D. Wagner, "Are you ready to lock?" in *Proc. 21st ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 750–761.
[28] D. Crocker, T. Hansen, and M. Kucherawy, "Domainkeys identified mail (DKIM) signatures," RFC 6376, Sep. 2011.
[29] Z. Durumeric *et al.*, "Neither snow nor rain nor MITM...: An empirical analysis of email delivery security," in *Proc. ACM Conf. Internet Meas. Conf.*, 2015, pp. 27–39.
[30] Avispa project: Avispa web tool. [Online]. Available: http://www.avispa-project.org/

[31] Amazon web service. [Online]. Available: https://aws.amazon.com/
[32] Postfix. [Online]. Available: http://www.postfix.org/
[33] opendkim. [Online]. Available: http://www.opendkim.org/
[34] Enron corpus. [Online]. Available: http://www.cs.cmu.edu/enron/
[35] J. Bonneau, S. Preibusch, and R. Anderson, "A birthday present every eleven wallets? The security of customer-chosen banking pins," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2012, pp. 25–40.
[36] R. Veras, C. Collins, and J. Thorpe, "On the semantic patterns of passwords and their security impact," in *Proc. 21th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2014.
[37] Y. Li, H. Wang, and K. Sun, "A study of personal information in human-chosen passwords and its security implications," in *Proc. 35th Annu. Int. Conf. Comput. Commun.*, 2016, pp. 1–9.
[38] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password memorability and security: Empirical results," *IEEE Security Privacy Mag.*, vol. 2, no. 5, pp. 25–31, Sep./Oct. 2004.
[39] C. Kuo, S. Romanosky, and L. F. Cranor, "Human selection of mnemonic phrase-based passwords," in *Proc. 2nd Symp. Usable Privacy Secur.*, 2006, pp. 67–78.
[40] J. H. Huh, S. Oh, H. Kim, K. Beznosov, A. Mohan, and S. R. Rajagopalan, "Surpass: System-initiated user-replaceable passwords," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 170–181.
[41] L. Wang, Y. Li, and K. Sun, "Amnesia: A bilateral generative password manager," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst.*, 2016, pp. 313–322.
[42] M. Just, "Designing and evaluating challenge-question systems," *IEEE Security Privacy Mag.*, vol. 2, no. 5, pp. 32–39, Sep./Oct. 2004.
[43] J. Bonneau and S. Preibusch, "The password thicket: Technical and market failures in human authentication on the web," in *Proc. 9th Workshop Econ. Inf. Secur.*, 2010.
[44] H. Sun, K. Sun, Y. Wang, and J. Jing, "TrustOTP: Transforming smartphones into secure one-time password tokens," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 976–988.
[45] I. Jermyn *et al.*, "The design and analysis of graphical passwords," in *Proc. 8th Conf. USENIX Secur. Symp.*, 1999, Art. no. 1.
[46] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE Security Privacy*, vol. 1, no. 2, pp. 33–42, Mar./Apr. 2003.
[47] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain, "Biometric cryptosystems: Issues and challenges," *Proc. IEEE*, vol. 92, no. 6, pp. 948–960, Jun. 2004.
[48] N. Zheng, A. Paloski, and H. Wang, "An efficient user verification system via mouse movements," in *Proc. 18th ACM SIGSAC Conf. Comput. Commun. Secur.*, 2011, pp. 139–150.
[49] Bank of america forgot passcode. [Online]. Available: https://secure.bankofamerica.com/login/reset/-entry/forgotPwdScreen.go
[50] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," in *Proc. 9th ACM SIGSAC Conf. Comput. Commun. Secur.*, 2002, pp. 161–170.
[51] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth-factor authentication: Somebody you know," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 168–178.
[52] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognit.*, vol. 37, pp. 2245–2255, 2004.
[53] F. Aloul, S. Zahidi, and W. El-Hajj, "Two factor authentication using mobile phones," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl.*, 2009, pp. 641–644.
[54] A. Czeskis, M. Dietz, T. Kohno, D. Wallach, and D. Balfanz, "Strengthening user authentication through opportunistic cryptographic identity assertions," in *Proc. 19th ACM Conf. Comput. Commun. Secur.*, 2012, pp. 404–414.
[55] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Soundproof: Usable two-factor authentication based on ambient sound," in *Proc. 24th USENIX Conf. Secur. Symp.*, 2015, pp. 483–498.
[56] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos, "Progressive authentication: Deciding when to authenticate on mobile phones," in *Proc. 21st USENIX Secur. Symp.*, 2012, pp. 301–316.

Yue Li received the BEng degree from the Information Engineering Department, Chinese University of Hong Kong, Hong Kong, in 2013, and the PhD degree from the Computer Science Department, College of William and Mary, Williamsburg, Virginia, in 2019. His research interests include secure authentication, network security, and attack forensics analysis. After graduation, he joined the Facebook, Inc., and is currently working on highly scalable and dependable machine learning infrastructures.

Zeyu Chen received the BSc degree in physics from the University of Science and Technology of China, Hefei, China, in 2015, and the MEng degree from the Department of Electrical and Computer Engineering, University of Delaware, Newark, Delaware, in 2017. He is currently working toward the PhD degree in the Department of Electrical and Computer Engineering, University of Delaware, Newark, Delaware. His research interests inlcude secure authentication and defense against memory corruptions in software.

Haining Wang received the PhD degree in computer science and engineering from the University of Michigan, Ann Arbor, Michigan, in 2003. He is currently a professor of electrical and computer engineering with Virginia Polytechnic Institute and State University, Arlington, Virginia. Before that, he was a professor of ECE with the University of Delaware. His research interests include areas of security, networking system, and cloud computing. He has published more than 150 technical papers in refereed journals and conference proceedings.

Kun Sun received the PhD degree in computer science from North Carolina State University, Raleigh, North Carolina, in 2006. Currently, he is an associate professor with the Department of Information Sciences and Technology, George Mason University. He is also the director of Sun Security Laboratory. Before joining GMU, he was an assistant professor with the College of William and Mary. His research focuses on systems and network security. He has more than 15 years working experience in both industry and academia, publishing more than 80 conference and journal papers. His current research interests include trustworthy computing environment, moving target defense, software security, password management, and software defined networking.

Sushil Jajodia is university professor, BDM international professor, and director of Center for Secure Information Systems, George Mason University. He has an extensive track record in cyber security research and significant leadership experience, having served in various leadership roles throughout his careers. He has authored or coauthored seven books, edited 52 books and conference proceedings, and published more than 500 technical papers in refereed journals and conference proceedings. He holds 23 U.S. patents, and has received a number of prestigious awards in recognition of his research accomplishments. According to the Google Scholar, he has more than 43 000 citations and his h-index is 104.